

Exact computation of joint spectral characteristics of linear operators

Nicola Guglielmi · Vladimir Protasov

Abstract We address the problem of the exact computation of two joint spectral characteristics of a family of linear operators, the joint spectral radius (in short JSR) and the lower spectral radius (in short LSR), which are well-known different generalizations to a set of operators of the usual spectral radius of a linear operator. In this article we develop a method which - under suitable assumptions - allows to compute the JSR and the LSR of a finite family of matrices exactly. We remark that so far no algorithm was available in the literature to compute the LSR exactly.

The paper presents necessary theoretical results on extremal norms (and on extremal antinorms) of linear operators, which constitute the basic tools of our procedures, and a detailed description of the corresponding algorithms for the computation of the JSR and LSR (the last one restricted to families sharing an invariant cone). The algorithms are easily implemented and their descriptions are short.

If the algorithms terminate in finite time, then they construct an extremal norm (in the JSR case) or antinorm (in the LSR case) and find their exact values; otherwise they provide upper and lower bounds that both converge to the exact values. A theoretical criterion for termination in finite time is also derived. According to numerical experiments, the algorithm for the JSR finds the exact value for the vast majority of matrix families in dimensions ≤ 20 . For nonnegative matrices it works faster and finds JSR in dimensions of order 100 within a few iterations; the same is observed for the algorithm computing the LSR. To illustrate

The research of the first author is supported by Italian M.I.U.R. (PRIN2007 project) and G.N.C.S.

The research of the second author is supported by the RFBR grants No 11-01-00329 and No 10-01-00293, and by the grant of Dynasty foundation.

Nicola Guglielmi
Department of Pure and Applied Mathematics,
University of L'Aquila, Italy
E-mail: guglielm@univaq.it

Vladimir Protasov
Department of Mechanics and Mathematics, Moscow State University, Vorobyovy Gory,
Moscow, Russia 119992
E-mail: v-protasov@yandex.ru

the efficiency of the new method we are able to apply it in order to give answers to several conjectures which have been recently stated in combinatorics, number theory, and the theory of formal languages.

Keywords Linear operator · joint spectral radius · lower spectral radius · algorithm · polytope · extremal norm · antinorm.

AMS 2010 subject classification: 15A60, 15-04, 15A18, 90C90

1 Introduction and background

The joint spectral characteristics of linear operators are now applied in many areas, from functional analysis and dynamical systems to discrete mathematics and number theory. We focus on two characteristics: the joint spectral radius and the lower spectral radius, and elaborate a method of their exact computation applicable even for relatively high dimensions.

The joint spectral radius of a set of matrices is a measure identifying the highest possible rate of growth of the norm of products of matrices (with no ordering and with repetition permitted) of the set. In contraposition, the lower spectral radius defines the lowest possible rate of growth. Both measures appear in several applications (see e.g. Strang [Str]).

In this paper we consider the problem of the computation of both joint spectral characteristics for a finite set of matrices. In contrast to the fact that in the last twenty years much effort has been devoted to the computation of the joint spectral radius, very little is known about computing the lower spectral radius (to the best of our knowledge, the only available method of its approximate computation was presented in [PJB]).

The joint spectral radius originated with Rota and Strang in 1960 [RS], and became extremely popular after Daubechies and Lagarias [DL] revealed its role in the study of refinement equations and wavelets. Since then it has found applications in functional equations, approximation, probability, combinatorics, etc. (see [J,PJB] for the extensive bibliography). Let

$$\mathcal{M} = \{A_1, \dots, A_m\}$$

be a finite family of linear operators acting in \mathbb{R}^d . We write

$$\mathcal{M}^k = \{A_{d_k} \dots A_{d_1} \mid d_j \in \{1, \dots, m\}, j = 1, \dots, k\}$$

for the set of all m^k products of length k of operators from \mathcal{M} . The *joint spectral radius* (JSR) of the family \mathcal{M} is

$$\widehat{\rho}(\mathcal{M}) = \lim_{k \rightarrow \infty} \max_{B \in \mathcal{M}^k} \|B\|^{1/k}. \quad (1)$$

This limit exists for every family \mathcal{M} and does not depend on the norm in \mathbb{R}^d [BW]. Clearly, if \mathcal{M} consists of one operator A_1 , then $\widehat{\rho}(\mathcal{M}) = \rho(A_1)$, where $\rho(A_1)$ is the (usual) spectral radius of A_1 , which is the maximal modulus of its eigenvalues.

For any family \mathcal{M} there is a positive constant c_1 such that

$$\max_{d_1, \dots, d_k} \|A_{d_k} \dots A_{d_1}\| \geq c_1 \widehat{\rho}^k$$

for every $k \in \mathbb{N}$. The family is called *non-defective* if the inverse estimate holds, that is there is a constant c_2 such that $\max_{d_1, \dots, d_k} \|A_{d_k} \dots A_{d_1}\| \leq c_2 \hat{\rho}^k$. It appears that if a family \mathcal{M} is irreducible, i.e., its operators do not share a common nontrivial invariant subspace of \mathbb{R}^d , then it is non-defective [P1]. Thus, for an irreducible family one has $\max_{d_1, \dots, d_k} \|A_{d_k} \dots A_{d_1}\| \asymp \hat{\rho}^k$, where the symbol \asymp denotes asymptotic equivalence ($a_k \asymp b_k$ if there are constants $c_1, c_2 > 0$ such that $c_1 a_k \leq b_k \leq c_2 a_k$ for all k). Whence, the joint spectral radius is the exponent of polynomial growth for the largest norm of operator products of length k . The geometric sense of JSR is the following: $\hat{\rho} < 1$ if and only if there exists a norm in \mathbb{R}^d such that $\|A_j\| < 1$ for all $j = 1, \dots, m$, where $\|A\| = \sup_{\|x\| \leq 1} \|Ax\|$ is the corresponding operator norm. In other words, $\hat{\rho} < 1$ precisely when there is a norm in \mathbb{R}^d with respect to which all operators from \mathcal{M} are contractive. So, it is natural to expect that each family of operators possesses some special norms related to JSR. The following theorem established in 1988 by Barabanov [B] shows that this is indeed the case, at least for irreducible families. A norm $\|\cdot\|$ in \mathbb{R}^d is called *invariant* for \mathcal{M} if there is a number $\lambda \geq 0$ such that $\max_{j=1, \dots, m} \|A_j x\| = \lambda \|x\|$ for every $x \in \mathbb{R}^d$. It is shown easily that for every invariant norm one has $\lambda = \hat{\rho}(\mathcal{M})$.

Theorem 1 [B] *Every irreducible family \mathcal{M} possesses an invariant norm.*

In practice it suffices to get a special norm with some weaker requirements, the so-called *extremal norm*.

Definition 1 *A norm $\|\cdot\|$ is called extremal for \mathcal{M} if $\|A_j x\| \leq \hat{\rho} \|x\|$ for all $x \in \mathbb{R}^d$.*

Thus, the norm is extremal if and only if $\max_{j=1, \dots, m} \|A_j\| = \hat{\rho}$. Indeed, from the definition it follows that $\max_{j=1, \dots, m} \|A_j\| \leq \hat{\rho}$; on the other hand, the submultiplicativity of operator norms yields $\max_{j=1, \dots, m} \|A_j\| \geq \hat{\rho}$. Whence, this inequality becomes an equality precisely for extremal norms. This property justifies the term “extremal”.

Clearly, any invariant norm is extremal, but not vice versa. Let us remark that for every extremal norm one has $\max_{d_1, \dots, d_k} \|A_{d_k} \dots A_{d_1}\| = \hat{\rho}^k$, $k \in \mathbb{N}$, i.e., the asymptotic equality becomes a sharp equality for all k . In particular, for $k = 1$ we have $\max_j \|A_j\| = \hat{\rho}$. Thus, if we know an extremal norm, then we have the exact value of JSR. The main idea of the approach presented in this paper is to find JSR by constructing (in an iterative way) an extremal norm.

We first present an algorithm for general sets of matrices, which under some suitable assumptions is able to check if a certain product in the multiplicative semigroup is spectrum maximizing. The algorithm is based on the computation of an extremal norm whose unit ball is a balanced polytope, and we provide it by a new criterion assuring a finite time termination and a new stopping condition.

Then we analyze sets of matrices having an invariant cone (a most important case is given by families of nonnegative matrices). For such sets we refine the algorithm for the general case and exploit the invariant property of the set in order to make the algorithm faster. Under this assumption we are also able to determine an algorithm for the exact computation of the lower spectral radius, which appears to be the first algorithm able to provide an exact value of this important measure.

The algorithms compute respectively a bounded and an unbounded polytope which represent the unit balls of, respectively, an extremal norm and antinorm for the considered set.

We write the formal routines of the algorithms and illustrate their efficiency by suitable examples and by numerical tests with randomly generated matrices. As we shall see, the algorithms find the exact value of JSR for general families of matrices (under some minor restrictions) in dimensions up to 20. For nonnegative matrices they work surprisingly fast even in dimension $d = 100$ and higher. Let us remark that our approach does not apply successfully to all families. There are cases, in fact, where the algorithms we propose are not able to finitely compute the exact values of the considered joint spectral characteristics, but only to approximate them.

In view of negative complexity results for the problem of JSR computation [BT], this is unlikely that there are effective methods applicable for all families of operators. Nevertheless, we claim that our approach works for the vast majority of families. The results of many numerical tests with randomly generated matrices of dimension from 5 to 100 (some of them are presented at the end of this paper) confirm this claim. In all the cases the algorithms found the exact values of JSR. As a further confirmation of this, we are able to apply the new method to solve several open problems in combinatorics and discrete mathematics.

In the literature there are several methods for the computation of the JSR. Some of them work only for small dimensions d , but give either an exact or a very accurate value of $\hat{\rho}$. For example, the method of polytope norms [P1,GZ1,BJP,GZ2,CGSZ]; see also special methods in [V,P3,HMR] elaborated for particular matrices.

Other methods aim to an approximate computation, such as the Kronecker lifting method [P2,BN], ellipsoidal norm method [BNT], Gripenberg's branch-and-bound method [DL,G] can work for bigger dimensions (mostly, up to 20), but produce pretty rough estimations. Recent approaches involving some modern tools of convex optimization (conic and semidefinite programming, sum-of-squares approximation, etc.) have rather good accuracy for higher dimensions (10 or even bigger) [PJ,PJB]. Most of those methods are actually based on the same simple idea. For each k we have

$$\max_{B \in \mathcal{M}^k} [\rho(B)]^{1/k} \leq \hat{\rho}(\mathcal{M}) \leq \max_{B \in \mathcal{M}^k} \|B\|^{1/k}. \quad (2)$$

The right hand side of this inequality converges to $\hat{\rho}$ as $k \rightarrow \infty$, which follows from the definition. The upper limit of the left hand side also equals to $\hat{\rho}$ [BW]. So, choosing k large enough, it is possible to approximate JSR as close as we need. However, this possibility is purely theoretical, because in most of practical cases the number k grows as $\frac{C}{\varepsilon}$, where $\varepsilon > 0$ is the relative accuracy of the JSR approximation, and $C > 0$ is a constant, which may be large for high dimensions d . That is why the number of matrix products of \mathcal{M}^k to look over becomes enormous. The reason is that the norm $\|\cdot\|$ in the right hand side of (2) may not suit our family \mathcal{M} , i.e., it may be far from the extremal norm of that family. That is why, to achieve a good approximation of JSR one needs to find an appropriate norm in \mathbb{R}^d for the right hand side of (2). Actually, all the methods of JSR computation use various techniques to find such a special norm for a given family \mathcal{M} . Those are, for instance, a polytope norm [P1,GZ1], an ellipsoidal norm [BNT], a norm generated by a cone [PJB], a norm defined by a sum-of-squares polynomial [PJ], etc. Sometimes this idea leads even to finding the precise values of JSR. This

happens when both the inequalities in (2) become equalities. If we write

$$\rho_k = \max_{B \in \mathcal{M}^s, s \leq k} [\rho(B)]^{1/s},$$

then for the extremal norm we have $\|A\| \leq \rho_k$ for all $A \in \mathcal{M}$, and therefore $\rho_k = \hat{\rho}$. For example, if all the matrices of \mathcal{M} are symmetric, then the Euclidean norm is extremal, if they are all column-stochastic, then the L_1 -norm is so. In some practical cases people succeed in finding extremal norms for concrete pairs of matrices arising in various applications: Gripenberg [G] (matrices of Daubechies wavelets, dimensions from 4 to 7), Hechler, Mößner, and Reif [HMR] (matrices of the four-point subdivision schemes, $d = 4$), Protasov [P4] (de Rham matrices, $d = 2$), Guglielmi, Wirth, and Zennaro [GWZ] (matrices of the Blondel-Theys-Vladimiov family, $d = 2$), Protasov [P3] (matrices of the binary partition function, $d = 4, \dots, 12$), Villemoes [V] (matrices of refinement equations, $d = 2$), Guglielmi, Manni and Vitale [GMV] (matrices in Hermite subdivision schemes) etc. The JSR computation in each case was a nontrivial problem and required special tricks applicable only for some narrow classes of matrices.

The method of exact JSR computation presented in this paper is related to previous works (see [P1, GZ1, BJP, GZ2, CGSZ]) and aims to develop further ideas both for the general case, which we are going to recall, and for certain specific important cases, like that of nonnegative matrices. The method is applicable for all families of matrices, under some general assumptions. The main idea proposed in the above mentioned papers is to build an extremal norm, whose unit sphere is a polytope. At the first step we look over all products of matrices from \mathcal{M} of length at most l , and find a product Π , for which the value $[\rho(\Pi)]^{1/n}$ is maximal (n is the length of Π). Then we denote this value by ρ_l and try to prove that $\hat{\rho}(\mathcal{M}) = \rho_l$.

Definition 2 A product $\Pi \in \mathcal{M}^n$ is a *spectrum maximizing product (s.m.p.)* if

$$[\rho(\Pi)]^{1/n} = \hat{\rho}(\mathcal{M}).$$

To prove that Π is an s.m.p. it suffices to have an extremal norm $\|\cdot\|$ in \mathbb{R}^d , for which $\|A_j\| \leq \rho_l$, $j = 1, \dots, m$. By (2) in this case we indeed have $\rho_l = \hat{\rho}$. We try to build a *polytope extremal norm*, whose unit sphere is some polytope P . Such a polytope will also be called *extremal*. It is characterized by the property $A_j P \subset \rho_l P$, $j = 1, \dots, m$. The polytope is constructed successively: its first vertices are the *leading eigenvector* v_1 of Π (i.e., the eigenvector corresponding to the largest by modulo eigenvalue, which is assumed to be real for the moment), the leading eigenvectors v_i of the $(n-1)$ cyclic permutations of Π , and the same vectors taken with minus, i.e., $-v_i$. We call an eigenvalue λ of an operator A *leading* if $|\lambda| = \rho(A)$.

Then we consider their images $(\rho_l)^{-1} A_j v_i$, $j = 1, \dots, m$ and remove those are in the convex hull of the previous ones, etc., until we obtain a set of points \mathcal{V} such that

$$(\rho_l)^{-1} A_j \mathcal{V} \subset \text{co}_s(\mathcal{V}), \quad j = 1, \dots, m.$$

By $\text{co}_s(\mathcal{V})$ we denote the *symmetrized convex hull*: $\text{co}_s(\mathcal{V}) = \text{co}(\mathcal{V} \cup (-\mathcal{V}))$, where $\text{co}(\cdot)$ is the (usual) convex hull. Then the polytope $P = \text{co}_s(\mathcal{V})$ possesses the

desired property: $(\rho_l)^{-1}A_jP \subset P$, so P is an extremal polytope. This implies $\hat{\rho} = \rho_l$. The algorithm involves standard tools of linear programming.

In case the leading eigenvalue of Π is complex, one has to replace polytopes by the so-called *complex polytopes* (see e.g. [GZ3]).

Our goal is to develop this approach for general families of matrices in higher dimensions, to analyze the structure of extremal polytopes and to derive the conditions of convergence of this algorithm.

Let us now emphasize the shortcomings of our approach. First of all, not every family of matrices has an s.m.p. Moreover, even if a non-defective family \mathcal{M} possesses an s.m.p., it may not have extremal polytopes (neither real nor complex [JP]). For such families our method apparently does not work. Another disadvantage appears, when the s.m.p. is not unique, up to cyclic permutations. In this case an extremal polytope, even if it exists, in general cannot be found by our method. The first two cases are rather pathological. It required constructing special nontrivial examples to show that they are possible [BTV,JP]. The third case of multiple s.m.p., in contrast, being also quite rare in general, nevertheless, appears in practical applications.

We believe that our method can be extended to this case as well, which may be a challenging problem for further research.

To work with those “bad cases”, we apply our approach also to approximate computation of JSR. The algorithm constructs a polytope, which is either extremal or not. If it is, then the JSR is found. Otherwise, we stop the algorithm after a certain iteration, say the N -th, and use the obtained polytope as a unit ball of the corresponding norm in estimations (2). In most cases this gives very sharp bounds for JSR. Thus, for an arbitrary family \mathcal{M} the algorithm either produces an extremal polytope, or a polytope norm that gives good upper and lower bounds for JSR. Proposition 1 guarantees that both those bounds converge to $\hat{\rho}(\mathcal{M})$ as $N \rightarrow \infty$.

The second part of the paper deals with the *lower spectral radius* (LSR) defined as follows:

$$\check{\rho}(\mathcal{M}) = \lim_{k \rightarrow \infty} \min_{B \in \mathcal{M}^k} \|B\|^{1/k}. \quad (3)$$

Thus, LSR is the exponent of asymptotic growth of the *minimal* product of operators from the family \mathcal{M} . This notion defined in [Gu] have been applied in problems of dynamical systems, functional analysis, coding theory, combinatorics, number theory, etc. (see [J] for many references). The limit in (3) always exists and does not depend on the norm. A simple observation is that LSR can be estimated by the usual spectral radii as follows:

$$\check{\rho}(\mathcal{M}) \leq \min_{B \in \mathcal{M}^k} [\rho(B)]^{1/k} \leq \min_{B \in \mathcal{M}^k} \|B\|^{1/k}. \quad (4)$$

In contrast to inequality (2) for JSR, estimation (4) gives only upper bounds. In fact, there is no effective lower bounds for LSR, and this causes the main difficulty for its computation. Basically, the lower spectral radius is still harder to compute or to estimate than the joint spectral radius (see, for instance, [TB] for the corresponding complexity results). The notions of invariant and extremal norms cannot be directly extended to LSR. The reason is that the operation of taking minimum of several functions, in contrast to the maximum, does not obey convexity. This means that the pointwise minimum of several convex functions

may not be convex. Hence, the function $f(x) = \min_{j=1,\dots,m} \|A_j x\|$, $x \in \mathbb{R}^d$, in general, is not a norm in \mathbb{R}^d . To overcome this difficulty, we use in Section 6 a notion of *antinorm* defined on a convex cone $K \subset \mathbb{R}^d$ (Definition 4). This notion originated in [P6] to study the Lyapunov exponents of linear operators. As we shall see, it can also be applied to analyze the lower spectral radius. We prove that every family of operators that share a common invariant cone K possesses an extremal antinorm on that cone (Theorem 5). This allows us to extend the new approach to the LSR computation, replacing norms by antinorms, and polytopes by *infinite polytopes*, i.e. the sets of the type $\text{co}(\mathcal{V}) + K$, where $\mathcal{V} \subset \mathbb{R}^d$ is a finite set, and $K \subset \mathbb{R}^d$ is a cone. In particular, this approach can be used for nonnegative matrices, since the corresponding operators preserve the cone $K = \mathbb{R}_+^d$. This yields an algorithm of exact computation of LSR for nonnegative matrices. In numerical examples we show that the algorithm works well for rather big dimensions (like $d = 100$). Let us note that the problem of LSR computation for nonnegative matrices arise naturally in combinatorics, discrete mathematics, and number theory [C,P3,JPB2]. Some of those applications will be considered in detail in Section 9. See also [MS,FV] for applications to the problem of stabilization of switched linear systems.

The main results of the paper can be summarized as follows:

- (i) we analyze the considered algorithm for the JSR computation of an arbitrary family and improve it by elaborating a stopping criterion that indicates whether a chosen product Π can be an s.m.p. or not. If our initial guess is wrong, and Π is not an s.m.p., then the criterion determines it (usually, after a few iterations) and suggests a new candidate for s.m.p. with a bigger spectral radius.
- (ii) Theorem 4 in Section 5 gives a criterion for a family \mathcal{M} insuring that the algorithm terminates within finite time, i.e., produces an extremal polytope.
- (iii) we improve the considered algorithm when applied to nonnegative matrices; the new algorithm finds the exact values of JSR in much higher dimensions (up to $d = 100$);
- (iv) we obtain a new algorithm which is able to exactly compute the LSR for families of nonnegative matrices, by computing a polytope extremal antinorm;
- (v) as examples we compute the exact values of JSR for special families of matrices (of dimensions up to 40) from well-known problems of combinatorics and number theory. This, in particular, allows us to solve three open problems. We discuss this aspect below in more detail.
- (vi) we provide numerical tests with randomly generated matrices (both arbitrary and nonnegative), showing that for all considered cases the algorithms produce extremal polytopes and, consequently, the exact value of the JSR (LSR).

The structure of the paper is the following. We describe the algorithm for JSR computation in three possible cases, which will be considered separately and called **(R)**, **(C)** and **(P)**. The case **(R)**, when the leading eigenvalue of the product $\Pi \in \mathcal{M}^n$ (a candidate for s.m.p.) is real, is recalled and further analyzed in Section 2. We discuss an algorithm for constructing an extremal polytope and for computing JSR, give necessary explanations and proofs, and establish two efficiency results: on the stopping criterion (to indicate within finite time, whether

the chosen product Π is s.m.p. or not) and on the estimation for JSR. Thus, Algorithm **(R)** either terminates within finite time, in which case JSR is found, or produces lower and upper bounds converging to JSR. According to our numerical experiments with randomly generated matrices (Section 9), for almost all matrix families Algorithm **(R)** finds the exact value of JSR, and works efficiently for dimensions up to 20.

In Section 3 we briefly consider the case **(C)**, when the leading eigenvalue of Π is complex, where we refer to [GWZ, GZ2, GZ3]. The algorithm and all the efficiency results are very similar, but with complex polytopes. By the numerical results in Section 9, it works slower than Algorithm **(R)**, and works in smaller dimensions.

In Section 4 we consider the case **(P)**, when all matrices are nonnegative. In this case the corresponding Algorithm **(P)** works faster and much more efficiently. Of course, **(P)** is a special case of **(R)**, which is, in turn, a special case of **(C)**. In fact all the three algorithms are very similar and differ in a few key details. Nevertheless, we describe them separately and independently of each other for convenience of the reader. Besides, their practical efficiency is very different, and it would be non-reasonable to compute JSR of nonnegative matrices by Algorithm **(R)** or by **(C)**.

In Section 5 we formulate one of the main results of the paper. This is a criterion of terminating of Algorithms **(R)**, **(C)**, and **(P)** within finite time (Theorem 4). It shows that an algorithm produces an extremal polytope and finds the precise values of JSR if and only if the family \mathcal{M} has a *dominant product* (Definition 3). In particular, if the algorithm terminates within finite time, then Π is a dominant product for \mathcal{M} .

In Section 6 we extend our method to the lower spectral radius computation. To this end we first define an antinorm, prove several theoretical results about it, and then describe Algorithm **(L)** for the exact computation of LSR of nonnegative matrices. Its practical efficiency for randomly generated matrices (Section 9) is approximately the same as for Algorithm **(P)**.

Section 7 presents two detailed examples in dimension 2 to illustrate the algorithms.

In Section 8 we consider applications to several problems of combinatorics, coding theory and number theory. In the problem of asymptotic growth of the number of overlap-free words (§8.1) we compute precise values of exponents of the upper and lower growth. This proves two conjectures stated in 2008 [JPB2]. Then in §8.2 we do the same for the problem of density of ones in the Pascal rhombus, and disprove one previously known conjecture. In §8.3 and §8.4 we find precise values of the lower and upper growths of the Euler partition functions for some values of the parameters.

Section 9 presents the results of numerical tests for JSR and LSR computation for randomly generated matrices of dimensions from 5 to 100. In all the cases the algorithms find the exact values of JSR and LSR, which suggests that our approach generically has finite convergence.

In the sequel we assume that the basis $\{e_i\}_{i=1}^d$ of the space \mathbb{R}^d is fixed and do not distinguish between operators and the corresponding matrices. An eigenvalue λ is simple, if it is of multiplicity 1. The largest by modulo eigenvalue of an operator B is called *leading* and denoted by λ_{\max} (if there are several such eigenvalues, then

each of them is leading). We use the following notation: B^* is the operator adjoint to B , $\text{int } M$ is the interior of a set M , $\text{co}(M)$ is the convex hull of M . We use the short abbreviation “LP” for linear programming problems.

2 Computing of the joint spectral radius: the case of real leading eigenvectors (R)

In this section we present Algorithm (R) for JSR computation.

We consider an irreducible family $\mathcal{M} = \{A_1, \dots, A_m\}$. For some (as large as possible) l we look over all products Π of length $\leq l$ and take one with the biggest value $[\rho(\Pi)]^{1/n}$, where n is the length of the product. We denote it as $\Pi = A_{d_n} \cdots A_{d_1}$.

Let $\tilde{\mathcal{M}} = \{\tilde{A}_1, \dots, \tilde{A}_m\}$ be the normalized family, where $\tilde{A}_i = [\rho(\Pi)]^{-1/n} A_i$. For the product $\tilde{\Pi} = \tilde{A}_{d_n} \cdots \tilde{A}_{d_1}$ we have $\rho(\tilde{\Pi}) = 1$ which implies $\hat{\rho}(\tilde{\mathcal{M}}) \geq 1$.

Define, for an arbitrary nonzero vector $v \in \mathbb{R}^d$ the set

$$\Omega(v) = \bigcup_{k \geq 0} \left\{ \Gamma v \mid \Gamma \in \tilde{\mathcal{M}}^k \right\}, \quad (5)$$

(where $\tilde{\mathcal{M}}^0 = Id$, the identity matrix), i.e. the set obtained by joining v to all vectors obtained by applying the products of the semigroup of $\tilde{\mathcal{M}}$ to v . The following theorem (see [P1] and [GZ1]) relates the set $\Omega(v)$ and an extremal norm for $\tilde{\mathcal{M}}$.

Theorem 2 *Let $\tilde{\mathcal{M}} = \{\tilde{A}_1, \dots, \tilde{A}_m\}$ be irreducible and such that $\hat{\rho}(\tilde{\mathcal{M}}) \geq 1$ and let $\Omega(v)$ (for a given $v \neq 0$) be a bounded subset of \mathbb{R}^d spanning \mathbb{R}^d . Then $\hat{\rho}(\mathcal{M}) = 1$. Furthermore the set*

$$\overline{\text{co}_s(\Omega(v))} = \overline{\text{co}(\Omega(v) \cup -\Omega(v))} \quad (6)$$

is the unit ball of an extremal norm $\|\cdot\|$ for $\tilde{\mathcal{M}}$ (and for \mathcal{M}).

The main idea of the algorithm we present is to finitely compute the set (6) whenever it is a polytope. Let us clarify this key point.

We say that a bounded set $P \subset \mathbb{R}^d$ is a *balanced real polytope* (b.r.p.) if there exists a finite set of vectors $\mathcal{V} = \{v_i\}_{1 \leq i \leq p}$ (with $p \geq d$) such that $\text{span}(\mathcal{V}) = \mathbb{R}^d$ and

$$P = \text{co}_s(\mathcal{V}) = \text{co}(\mathcal{V}, -\mathcal{V}). \quad (7)$$

Therefore

$$P = \left\{ z = \sum_{x \in \mathcal{V}} t_x x \quad \text{with} \quad -q_x \leq t_x \leq q_x, \quad q_x \geq 0 \quad \forall x \in \mathcal{V} \quad \text{and} \quad \sum_{x \in \mathcal{V}} q_x \leq 1 \right\}.$$

The set P is the unit ball of a norm $\|\cdot\|_P$ on \mathbb{R}^d , which we call a *real polytope norm*.

Assume that the hypotheses of Theorem 2 hold. The possibility of actually determining an extremal polytope norm, if any, crucially relies on the search of the initial vector v , which we will address later in Theorem 4 that suggests to choose v as a leading eigenvector of Π (although a different choice would be admissible). We will assume here v to be real.

The idea is that of computing the set $\Omega(v)$ by applying recursively the family \widetilde{M} to a finite set of vectors (which in the beginning is simply the vector v), checking at every iteration h whether \widetilde{M} maps the symmetrized convex hull ($\text{co}_s(\Omega^{h-1}(v))$) of the computed set of vectors

$$\Omega^{h-1}(v) = \bigcup_{0 \leq k \leq h-1} \left\{ \Gamma v \mid \Gamma \in \widetilde{M}^k \right\},$$

into itself.

Algorithm **(R)** we are going to present is similar to the one described in [GZ1]; the main differences are that a new vertex is included even if it lies on the boundary of the current polytope, all the leading eigenvectors are considered as starting vertices of the searched extremal polytope and a new and efficient stopping criterion is added.

We start by an auxiliary result and then describe the algorithm.

Lemma 1 *Let an operator B have a unique simple leading eigenvalue $\lambda \in \mathbb{R}$ with the leading eigenvector v ; let also v^* be the leading eigenvector of B^* such that $(v^*, v) = 1$. If for some operator C one has $|(v^*, Cv)| > 1$, then for sufficiently large r the operator $B^r C$ has a unique simple leading eigenvalue, which is real and bigger than λ by modulo.*

Proof. Without loss of generality, after a suitable normalization, it can be assumed that $\lambda = 1$. Since all other eigenvalues of B are smaller by modulo than 1, it follows that B^r converges to the one-rank operator $B_\infty x = (v^*, x)v$ as $r \rightarrow \infty$. Hence $B^r C$ converges to the operator $B_\infty C$, whose unique simple leading eigenvalue is (v^*, Cv) , which exceeds 1 by modulo. \square

Remark 1 If v and v^* are the leading eigenvectors of B and B^* respectively, then these vectors cannot be orthogonal, otherwise the leading eigenvalue is not simple. So, $(v^*, v) \neq 0$, and hence, after a suitable normalization it can always be assumed that $(v^*, v) = 1$.

It is well-known that the problem of JSR computation has to be considered only for irreducible families of matrices, which do not possess common invariant linear subspaces. Otherwise this family is factorable in a suitable basis in \mathbb{R}^d : all the matrices A_j get a block upper-triangular form, and $\widehat{\rho}(\mathcal{M})$ equals to the maximal JSR of the blocks. This reduces the problem of JSR computation to several problems in smaller dimensions. Therefore, in the sequel of this section we assume that \mathcal{M} is irreducible.

2.1 Algorithm **(R)**

Initialization. Given the irreducible family $\mathcal{M} = \{A_1, \dots, A_m\}$ we look over all products Π of length $\leq l$ and consider the shortest product Π such that $[\rho(\Pi)]^{1/n}$ is maximal, where n is the length of the product. We denote it as $\Pi = A_{d_n} \cdots A_{d_1}$ and consider the main assumption:

- (i) *The product Π has a real nonzero leading eigenvalue.*

We assume that the leading eigenvalue λ_{\max} of Π is positive; the case of negative eigenvalue is considered in the same way. Let $\tilde{M} = \{\tilde{A}_1, \dots, \tilde{A}_m\}$ be the normalized family, where $\tilde{A}_i = [\rho(\Pi)]^{-1/n} A_i$. For the product $\tilde{\Pi} = \tilde{A}_{d_n} \cdots \tilde{A}_{d_1}$ we have $\lambda_{\max} = 1$.

Let $\tilde{\Pi}_1 = \tilde{\Pi}$, $\tilde{\Pi}_i = \tilde{A}_{d_{i-1}} \cdots \tilde{A}_{d_1} \tilde{A}_{d_n} \cdots \tilde{A}_{d_i}$ be a cyclic permutation of $\tilde{\Pi}_1$, $i = 2, \dots, n$. We denote by v_1 the leading eigenvector of $\tilde{\Pi}_1$, for which $\tilde{\Pi}_1 v_1 = v_1$. If it is not unique (in which case λ_{\max} is multiple) we take any of them. Then for every $i \geq 2$ we set

$$v_i = \tilde{A}_{d_{i-1}} \cdots \tilde{A}_{d_1} v_1.$$

Thus, v_i is a leading eigenvector of $\tilde{\Pi}_i$.

In case $\tilde{\Pi}$ has a unique simple eigenvalue, we also need the corresponding dual system of vectors: v_1^* the leading eigenvector of the conjugate operator $\tilde{\Pi}_1^*$ normalized by the condition $(v_1^*, v_1) = 1$ (see Remark 1), and

$$v_i^* = \tilde{A}_{d_i}^* \cdots \tilde{A}_{d_n}^* v_1^*$$

for $i = 2, \dots, n$. Thus, v_i^* is the leading eigenvector of $\tilde{\Pi}_i$, and $(v_i^*, v_i) = 1$. If the leading eigenvalue of $\tilde{\Pi}$ is multiple or not unique, then we do not need the conjugate system.

Set $k = 0$. We set $\mathcal{V}_0 = \mathcal{U}_0 = \{v_1, \dots, v_n\}$ and $\mathcal{R}_0 = \{(v_i, \tilde{A}_p) \mid i = 1, \dots, n; p = 1, \dots, m, p \neq d_i\}$.

Main loop

For $k \geq 1$. We have finite sets $\mathcal{V}_{k-1} \subset \mathbb{R}^d$, $\mathcal{U}_{k-1} \subset \mathcal{V}_{k-1}$, and $\mathcal{R}_{k-1} \subset \mathcal{U}_{k-1} \times \tilde{\mathcal{M}}$. Set $\mathcal{V}_k = \mathcal{V}_{k-1}$ and $\mathcal{U}_k = \emptyset$. Take an arbitrary pair $(v, \tilde{A}) \in \mathcal{R}_{k-1}$ and compute the norm whose unit ball is the polytope with vertices \mathcal{V}_{k-1} and $-\mathcal{V}_{k-1}$, of the corresponding vector $z = \tilde{A}v$. This is done by solving the following LP problem with variables $\{t_x\}_{x \in \mathcal{V}_k}$, $\{q_x\}_{x \in \mathcal{V}_k}$ and t_0 (which represents the reciprocal of the value of the norm):

$$\left\{ \begin{array}{ll} \max & t_0 \\ \text{subject to} & t_0 z = \sum_{x \in \mathcal{V}_k} t_x x \\ & -q_x \leq t_x \leq q_x, \quad q_x \geq 0 \quad \forall x \in \mathcal{V}_k \\ \text{and} & \sum_{x \in \mathcal{V}_k} q_x \leq 1, \end{array} \right. \quad (8)$$

The value of the problem, i.e., the value $\max t_0$ will be denoted by $t_{\{v, \tilde{A}\}}$. Thus, for a given pair $(v, \tilde{A}) \in \mathcal{R}_{k-1}$ we obtain the value $t_{\{v, \tilde{A}\}}$.

If $t_{\{v, \tilde{A}\}} > 1$, then we leave the sets \mathcal{V}_k and \mathcal{U}_k as they are, take the next pair $(v, \tilde{A}) \in \mathcal{R}_{k-1}$ and consider problem (8) for it.

If $t_{\{v, \tilde{A}\}} \leq 1$, then we distinguish between two cases

If the leading eigenvalue of Π is unique and simple, we apply the following

Stopping criterion:

For a given pair (v, \tilde{A}) we check the condition

$$|(v_j^*, \tilde{A}v)| \leq 1, \quad j = 1, \dots, n. \quad (9)$$

If (9) is satisfied, then we set $\mathcal{V}_k = \mathcal{V}_k \cup \{\tilde{A}v\}$, $\mathcal{U}_k = \mathcal{U}_k \cup \{\tilde{A}v\}$, take the next pair $(v, \tilde{A}) \in \mathcal{R}_{k-1}$ and consider problem (8) for it.

Otherwise If (9) is not satisfied, then Π is not an s.m.p. for \mathcal{M} , and $\hat{\rho}(\tilde{\mathcal{M}}) > 1$ (Lemma 1). We stop the algorithm and go either to the **Final step**, or back to the **Initialization**. In the latter case we need to find another candidate s.m.p. The first option is to increase l and to look over all products of a bigger length. Lemma 1 provides also a different approach. We take an index j , for which $|(v_j^*, \tilde{A}v)| > 1$. Applying Lemma 1 for the vectors v_j^* and $-v_j^*$, we conclude that there is r such that $\lambda_{\max}(\tilde{\Pi}_j^r \tilde{A}_{s_q} \cdots \tilde{A}_{s_1}) > 1$, where $\tilde{A}_{s_q} \cdots \tilde{A}_{s_1} v_j = \tilde{A}v$. We take the new initial product $\Pi = \Pi_j^r A_{s_q} \cdots A_{s_1}$ and restart the algorithm.

End If

Otherwise If the leading eigenvalue of Π is not unique or multiple, then we do not apply the stopping criterion, and set $\mathcal{V}_k = \mathcal{V}_k \cup \{\tilde{A}v\}$, $\mathcal{U}_k = \mathcal{U}_k \cup \{\tilde{A}v\}$, take the next pair $(v, \tilde{A}) \in \mathcal{R}_{k-1}$ and consider problem (8) for it.

End If

The k th step is over when the whole set \mathcal{R}_{k-1} is exhausted.

If $\mathcal{U}_k = \emptyset$, then $\hat{\rho}(\tilde{\mathcal{M}}) = 1$, and so $\hat{\rho}(\mathcal{M}) = [\rho(\Pi)]^{1/n}$. The extremal polytope is $P_{k-1} = \text{co}_s(\mathcal{V}_{k-1})$, and the s.m.p. for \mathcal{M} is Π . The algorithm terminates having performed k steps.

Otherwise If $\mathcal{U}_k \neq \emptyset$, then we set $\mathcal{R}_k = \mathcal{U}_k \times \tilde{\mathcal{M}}$ and continue.

End If

End For

Final step. If the algorithm has not terminated, then we stop it after some N steps, denote $t_N = \min_{(v, \tilde{A}) \in \mathcal{R}_{N-1}} t_{\{v, \tilde{A}\}}$, where $t_{\{v, \tilde{A}\}}$ is the solution of LP problem (8) for the last step, i.e., for $k = N$, and have the following estimate for the joint spectral radius of the family \mathcal{M} :

$$[\rho(\Pi)]^{1/n} \leq \hat{\rho}(\mathcal{M}) \leq t_N^{-1} [\rho(\Pi)]^{1/n}. \quad (10)$$

End of Algorithm (R).

Remark 2 An important difference with respect to previous similar algorithms is that if at step k a new vector v lies on the boundary of the polytope $P_{k-1} = \text{co}_s(\mathcal{V}_{k-1})$ (which means $t_{\{v, \tilde{A}_p\}} = 1$ for some p) then we include the vector as a new vertex. Clearly this condition is non generic and requires - to be tested in floating point arithmetics - the use of a suitable error tolerance.

Before we give the proofs, let us explain the general scheme of the algorithm.

2.2 The cyclic tree structure of the algorithm

Consider a combinatorial *cyclic tree* \mathcal{T} defined as follows. The root is formed by a cycle \mathbf{B} of n nodes v_1, \dots, v_n . They are, by definition, the nodes of zero level. For every $i \leq n$ an edge (all edges are directed) goes from v_i to v_{i+1} , where we set $v_{n+1} = v_1$. At each node of the root $m - 1$ edges start to nodes of the first level. So, there are $n(m - 1)$ different nodes on the first level. The sequel is by induction: there are $n(m - 1)m^{k-1}$ nodes of the k th level, $k \geq 1$, from each of them m edges (“children”) go to m different nodes of the $(k + 1)$ st level.

Consider now an arbitrary word $\mathbf{b} = d_n \dots d_1$ of length $n \geq 1$, where each d_j belongs to the alphabet $\{1, \dots, m\}$. The product of several words is their concatenation. We assume that \mathbf{b} is irreducible, i.e., is not a power of a shorter word. To every edge of the tree \mathcal{T} we associate a letter d as follows: the edge $v_i v_{i+1}$ corresponds to d_i , $i = 1, \dots, n$; at each node m edges start associated to m different letters. To a given word $q_k \dots q_1$ we associate the node, which is the end of the path from v_1 along the edges q_1, q_2, \dots, q_k . For example, the empty word corresponds to v_1 , the word \mathbf{b} also corresponds to v_1 , the word $d_2 d_1$ corresponds to v_3 , the word d_2 corresponds to either v_2 , if $d_2 = d_1$, or to a child of v_1 from the first level, otherwise. This tree is said to be generated by the word \mathbf{b} , or by the cycle \mathbf{B} .

For a family of operators $\tilde{\mathcal{M}} = \{\tilde{A}_1, \dots, \tilde{A}_m\}$ and for some product $\tilde{I} = \tilde{A}_{d_n} \dots \tilde{A}_{d_1}$ with an eigenvalue 1 we associate the cyclic tree \mathcal{T} generated by the word $d_n \dots d_1$. The node v_1 corresponds to an eigenvector with the eigenvalue 1; to a given node $v \in \mathcal{T}$ we associate a point $\tilde{A}_{q_k} \dots \tilde{A}_{q_1} v_1$, where the word $q_k \dots q_1$ corresponds to the node v .

When we start the algorithm, we take the set $\mathbf{B} = \{v_1, \dots, v_n\}$ as the root of the tree. At the first step we take any node v_i and consider successively its $(m - 1)$ children from the first level. For each neighbor $u = \tilde{A} v_i$, where $\tilde{A} \in \tilde{\mathcal{M}} \setminus \{\tilde{A}_{d_i}\}$ we solve LP problem (8) and determine, whether u belongs to the interior of the set $\text{co}_s(\mathcal{V}_1)$, where $\text{co}_s(M) = \text{co}\{M, -M\}$ is the symmetrized convex hull. If it does, then u is a “dead leaf” generating a “dead branch”: we will never come back to u , nor to nodes of the branch starting at u (so, this branch is cut off). If it does not, then u is an “alive leaf”, and we add this element u to the set \mathcal{V}_1 and to the set \mathcal{U}_1 . After the first step all alive leaves of the first level form the set \mathcal{U}_1 . At the second step we deal with the leaves from \mathcal{U}_1 only and obtain the next set of alive leaves of the second level \mathcal{U}_2 , etc. Thus, after the k th step we have a family \mathcal{U}_k of alive leaves from the k th level, and a set $\mathcal{V}_k = \cup_{j=0}^k \mathcal{U}_j$. A node u belongs to \mathcal{V}_k iff its level does not exceed k and it belongs to an alive branch starting from the root. The polytope P_k is the symmetrized convex hull $\text{co}_s(\mathcal{V}_k)$. The polytope P_{k-1} is extremal iff $\mathcal{U}_k = \emptyset$, i.e., the k th step produces no alive leaves (only dead ones). This means that there are no alive paths of length k from the root. Therefore $P_k = P_{k-1}$. Otherwise, if \mathcal{U}_k is nonempty, we make the next step and go to the $(k + 1)$ st level: take children of each element of \mathcal{U}_k , determine whether they are alive or dead and proceed.

2.3 Explanations and proofs

The algorithm produces a sequence of embedded polytopes $P_1 \subset P_2 \subset \dots$ such that $P_{j+1} = \text{co}_s \{ \tilde{A}_1 P_j, \dots, \tilde{A}_m P_j \}$ for every j . If the algorithm terminates after the k th step, then $P_k = P_{k-1}$. The k th step is actually needed only to ensure that the polytope P_{k-1} is extremal, i.e., $\tilde{A}_j P_{k-1} \subset P_{k-1}$, $j = 1, \dots, m$. In this case P_{k-1} possesses an interior of nonzero measure, otherwise its linear span is a common invariant nontrivial subspace of the family \tilde{M} , which contradicts the irreducibility assumption. Moreover, P_{k-1} is centrally-symmetric, hence $0 \in \text{int } P_{k-1}$. This, in particular, yields that if for some $v \in \mathbb{R}^d$ and $t > 1$ one has $tv \in P_{k-1}$, then $v \in \text{int } P_{k-1}$. Thus, if the value $t_{\{v, \tilde{A}\}}$ of LP problem (8) is bigger than 1, then $\tilde{A}v \in \text{int } P_{k-1}$. Thus all dead leaves removed by the algorithm are internal points for P_{k-1} .

In the Minkowski norm $\|\cdot\|_{k-1}$ whose unit ball is given by P_{k-1} , one has $\|\tilde{A}\|_{k-1} \leq 1$ for all $\tilde{A} \in \tilde{M}$, therefore $\hat{\rho}(\tilde{M}) \leq 1$. On the other hand, $\hat{\rho}(\tilde{M}) \geq \rho(\tilde{H})^{1/n} = 1$, hence $\hat{\rho}(\tilde{M}) = \rho(\tilde{H})^{1/n} = 1$, and so $\hat{\rho}(\mathcal{M}) = \rho(\Pi)^{1/n}$. Thus, if the algorithm terminates within finite time, then the s.m.p. and the exact value of JSR are found.

Suppose the algorithm does not terminate within finitely many steps. After the final step we take the polytope $P_{N-1} = \text{co}_s(\mathcal{V}_{N-1})$ as a unit ball of the new norm $\|\cdot\|_{N-1}$ in \mathbb{R}^d . Then $\max_{A \in \mathcal{M}} \|A\|_{N-1} \geq \hat{\rho}(\mathcal{M})$. We have

$$\max_{A \in \mathcal{M}} \|A\|_{N-1} = [\rho(\Pi)^{1/n}] \max_{\tilde{A} \in \tilde{M}} \|\tilde{A}\|_{N-1} = [\rho(\Pi)^{1/n}] \cdot \max_{v \in \mathcal{U}_{N-1}} (t_{\{v, \tilde{A}\}})^{-1},$$

where $t_{\{v, \tilde{A}\}}$ is the value of LP problem (8) for $k = N$. Therefore, $\hat{\rho}(\tilde{M}) \leq (t_N)^{-1}$, and after multiplying by $[\rho(\Pi)]^{1/n}$ we arrive at (10).

Remark 3 Although we perform operations numerically, the obtained results have to be considered exact since apart from the vertices of P_{k-1} , all other vectors obtained by applying the scaled matrices \tilde{A}_j to the vertices are either vertices or internal points to the polytope P_{k-1} .

Remark 4 By the construction of the algorithm, each vertex of the polytope P_k belongs either to \mathcal{V}_k or to $-\mathcal{V}_k$. However, not all elements of the set \mathcal{V}_k are actually vertices: some of them may lie in the convex hull of the others. This means that in general the set \mathcal{V}_k is not an essential system of vertices. Nevertheless, for the sake of simplicity we call all elements of \mathcal{V}_k vertices.

Remark 5 Actually Algorithm **(R)** can be applied to a reducible family \mathcal{M} as well. If the algorithm terminates after k th iteration, and the set $\mathcal{V}_k \subset \mathbb{R}^d$ does not lie in a linear subspace of a smaller dimension (i.e., the system of equations $(x, v) = 0$, $v \in \mathcal{V}_k$ has only trivial solution $x = 0$), then P_{k-1} is an extremal polytope, and Π is an s.m.p. Thus, one can apply Algorithm **(R)** without preliminary checking of irreducibility of \mathcal{M} . Nevertheless, if the family \mathcal{M} is reducible, then it is always better to factorize \mathcal{M} before applying Algorithm **(R)**, because this reduces the dimension of matrices.

2.4 Efficiency results for Algorithm (R)

If the algorithm terminates, then it finds the exact value of JSR, otherwise estimate (10) gives its approximate value with the relative error $\varepsilon = (t_N)^{-1} - 1$. This error depends on two integer parameters: the maximal length l of the products, among which we choose an s.m.p. Π , and the number of iterations N of the algorithm. Let us show that the error ε tends to zero as both these parameters increase:

Proposition 1 *For an arbitrary irreducible family \mathcal{M} we have $t_N \rightarrow 1$ as $l \rightarrow \infty$ and $N \rightarrow \infty$.*

Thus, both sides of inequality (10) tend to $\widehat{\rho}(\mathcal{M})$ as $l \rightarrow \infty$, $N \rightarrow \infty$. Algorithm (R) either finds the value of JSR or provides lower and upper bounds for it; those bounds are arbitrarily close to each other, whenever both l and N are large enough.

In the proof we use Dini's theorem on monotone convergence: if a sequence of continuous real-valued functions defined on a compact metric space Q is monotone and converges pointwise to a continuous function, then this convergence is uniform on Q (see [Ru, theorem 7.13]). We use the Minkowski norm $\|\cdot\|_D$ associated to a given symmetric convex body $D \subset \mathbb{R}^d$ as follows: $\|\cdot\|_D = \inf \{t^{-1} \mid t > 0, tx \in D\}$.

Proof of Proposition 1. Assume first that $\widehat{\rho}(\widetilde{\mathcal{M}}) = 1$, i.e., that Π is an s.m.p. The algorithm produces the polytopes $\{P_k\}_{k \in \mathbb{N}}$ such that

$$P_k \subset P_{k+1} = \text{co}_s \{\widetilde{A}_1 P_k, \dots, \widetilde{A}_m P_k\}.$$

Since the family $\widetilde{\mathcal{M}}$ is irreducible, there is $p \geq 1$ such that all the polytopes P_k have nonempty interior for $k \geq p$. Hence, for $k \geq p$ the polytope P_k generates the Minkowski norm $f_k(\cdot) = \|\cdot\|_{P_k}$. For each $x \in \mathbb{R}^d$ the sequence $\{f_k(x)\}_{k \geq p}$ is non-increasing. Moreover, it is uniformly bounded below by a positive constant, because all the polytopes $\{P_k\}_{k \geq p}$ are contained in some ball, since the family $\widetilde{\mathcal{M}}$ is non-defective. Therefore, the sequence $f_k(x)$ converges pointwise to a function $f(x)$, which is also a norm in \mathbb{R}^d . By Dini's theorem, this convergence is uniform on any compact subset of \mathbb{R}^d . In particular, it is on the unit sphere $S = \{x \in \mathbb{R}^d \mid f(x) = 1\}$ of the norm f . Thus, $f_k(x) \rightarrow 1$ uniformly for $x \in S$, as $k \rightarrow \infty$. Hence, there is N_ε such that $f_{N-1}(x) \leq 1 + \varepsilon$ for all $x \in S$, whenever $N \geq N_\varepsilon$. Consequently, $(t_N)^{-1} = \sup_{x \in P_N} f_{N-1}(x) \leq \sup_{x \in S} f_{N-1}(x) \leq 1 + \varepsilon$, which completes the proof for the case $\widehat{\rho} = 1$. Consider now the general case. We have $[\rho(\Pi)]^{1/n} \rightarrow \widehat{\rho}(\mathcal{M})$ as $l \rightarrow \infty$, where, let us remember, $n = n(l)$ is the length of Π . Hence, for every $\delta > 0$ there is l_δ such that $\widehat{\rho}(\widetilde{\mathcal{M}}) < 1 + \delta$. Since each polytope P_k continuously depends on the family $\widetilde{\mathcal{M}} = [\rho(\Pi)]^{-1/n} \mathcal{M}$, for all sufficiently small δ one has $(t_{N_\varepsilon})^{-1} < 1 + \varepsilon$. This inequality holds for all $l \geq l_\delta$. It remains to note that for every family \mathcal{M} the value t_N is non-decreasing in N . Indeed, $t_N P_N \subset P_{N-1}$, hence $t_N \widetilde{A}_j P_N \subset \widetilde{A}_j P_{N-1}$ for every $j = 1, \dots, m$, and so $t_N P_{N+1} \subset P_N$. Therefore, $t_{N+1} = \sup \{t > 0 \mid t P_{N+1} \subset P_N\} \geq t_N$. We see that $(t_N)^{-1} < 1 + \varepsilon$, provided $N \geq N_\varepsilon$. Thus, for every $\varepsilon > 0$ there are l_δ and N_ε such that $(t_N)^{-1} - 1 < \varepsilon$, whenever $N \geq N_\varepsilon$ and $l \geq l_\delta$. \square

Let us now show the efficiency of the stopping criterion.

Proposition 2 *Assume the leading eigenvalue of Π is real, unique and simple. If the assumption of the algorithm is wrong (i.e., Π is not an s.m.p.) then for every $j = 1, \dots, n$ condition (9) is violated at some step. Conversely, if condition (9) is violated at some step for some j , then Π is not an s.m.p.*

Proof. The sufficiency follows from Lemma 1. To prove the necessity suppose $\widehat{\rho}(\widetilde{M}) > 1$; then for every point $v \neq 0$ and for every number $R > 0$ there is a product C of operators of the family \widetilde{M} such that $\|Cv\| \geq R$ [P1, theorem 1]. On the other hand, since the family \widetilde{M} is irreducible, it follows that there is $\gamma > 0$ such that for every point $y \neq 0$ the set $K_d(y) = \text{co}_s \{Ay \mid A \in \widetilde{M}^d\}$ contains a ball of radius $\gamma\|y\|$ (see [K2] for the proof), where $\gamma > 0$ is a constant. Applying these results to the points $v = v_1$ and $y = Cv_1$ and using the fact that the polytope P_{s+d} contains $K_d(y)$, where s is the length of the product C , we see that the polytope P_{s+d} contains a ball of radius γR centered at the origin. Therefore,

$$\sup_{x \in P_{s+d}} |(v_j^*, x)| \geq \gamma R \|v_j^*\|.$$

Since this supremum is attained at some vertex of P_{s+d} , which is produced by the algorithm, we see that condition (9) will fail by the $(s+d)$ -th step, whenever $R > 1/(\gamma\|v_j\|^*)$. □

Thus, if the chosen product Π is not an s.m.p., then the stopping criterion always determines this in finite time. In Section 5 we formulate Theorem 4 that gives a sharp criterion for the algorithm to terminate in finitely many steps (and, respectively, to produce an extremal polytope).

3 Computing of the joint spectral radius: the case of complex leading eigenvectors (C)

For the theoretical results and the algorithms relevant to this case we mainly address the reader to the papers [GWZ, GZ2, GZ3].

We recall from [GZ3, VZ] the definition of a balanced complex polytope, which generalizes to the complex case a centrally symmetric real polytope.

Let $\mathcal{V} = \{v_i\}_{1 \leq i \leq p}$ be a finite set of vectors, then

$$\text{absco}(\mathcal{V}) = \left\{ z \in \mathbb{C}^d \mid z = \sum_{x \in \mathcal{V}} t_x x \text{ with } \sum_{x \in \mathcal{V}} |t_x| \leq 1 \right\}. \quad (11)$$

Definition 1 A set $\mathcal{P} \subset \mathbb{C}^d$ is a balanced complex polytope (b.c.p.) if there exists a finite set of vectors $\mathcal{V} = \{v_i\}_{1 \leq i \leq p}$ such that

$$\text{span}(\mathcal{V}) = \mathbb{C}^d \quad \text{and} \quad \mathcal{P} = \text{absco}(\mathcal{V}). \quad (12)$$

Moreover, if $\text{absco}(\mathcal{V}') \subsetneq \text{absco}(\mathcal{V})$ for all $\mathcal{V}' \subsetneq \mathcal{V}$, we say that \mathcal{V} is an essential system of vertices for \mathcal{P} . Every vector $u v_i$ with $u \in \mathbb{C}$, $|u| = 1$, is called a vertex of \mathcal{P} .

Note that geometrically a b.c.p. \mathcal{P} is not a classical polytope (see [GZ3]).

A polytope norm can be defined in a natural way.

Lemma 2 Any b.c.p. \mathcal{P} is the unit ball of a norm $\|\cdot\|_{\mathcal{P}}$ on \mathbb{C}^d .

The proof is immediate (see e.g. [GZ3])

Definition 2 We shall call *complex polytope norm* any norm $\|\cdot\|_{\mathcal{P}}$ whose unit ball is a b.c.p. \mathcal{P} .

The corresponding vector norm is characterized by the following Lemma (for a proof see [GZ3]).

Lemma 3 Let \mathcal{P} be a b.c.p. and let $\|\cdot\|_{\mathcal{P}}$ be the corresponding complex polytope norm. Then, for any $z \in \mathbb{C}^d$, it holds that

$$\|z\|_{\mathcal{P}} = \left\{ \max t_0 \mid t_0 z = \sum_{x \in \mathcal{V}} t_x x, \quad \sum_{x \in \mathcal{V}} |t_x| \leq 1 \right\}, \quad (13)$$

where $\mathcal{V} = \{v_i\}_{1 \leq i \leq p}$ is an essential system of vertices for \mathcal{P} .

Complex polytope norms are dense in the set of all norms defined on \mathbb{C}^d and consequently the corresponding set of induced matrix complex polytope norms is dense in the set of all induced $d \times d$ -matrix norms (see [GZ3]). This implies the following important property:

$$\hat{\rho}(\mathcal{M}) = \inf_{\|\cdot\|_{\mathcal{P}}} \max_{A \in \mathcal{M}} \|A\|_{\mathcal{P}}$$

where $\|\cdot\|_{\mathcal{P}}$ denotes the set of polytope norms.

From an algorithmic point of view, the above property has the consequence that although an extremal polytope norm may not exist, it is possible to compute a polytope norm which is ε -close to an extremal one, for any $\varepsilon > 0$.

3.1 Main differences between Algorithm (C) and Algorithm (R).

Algorithm **(R)** extends to the complex case in a direct way, as well as the convergence and approximation results (see [GWZ], [GZ2]). The only (important) difference lies in the computation of the polytope norm of a vector. We obtain this by rewriting (13) as a real optimization problem.

Let $\mathcal{P} = \text{absco}(\mathcal{V})$ (with $\mathcal{V} = \{v_1, v_2, \dots, v_p\}$) be a b.c.p. and $\|\cdot\|_{\mathcal{P}}$ the associated norm. For any $z \in \mathbb{C}^d$, we write (13) (with $t_x = \alpha_x + i\beta_x$) in the following way:

$$\left\{ \begin{array}{l} \max t_0 \\ \text{subject to } \sum_{x \in \mathcal{V}} \alpha_x \text{Re}(x) - \beta_x \text{Im}(x) = t_0 \text{Re}(z) \\ \sum_{x \in \mathcal{V}} \alpha_x \text{Im}(x) + \beta_x \text{Re}(x) = t_0 \text{Im}(z) \\ \text{and } \sum_{x \in \mathcal{V}} \sqrt{\alpha_x^2 + \beta_x^2} \leq 1 \end{array} \right. \quad (14)$$

This problem can be efficiently solved in the framework of the conic quadratic programming, by the interior point method on Lorentz cones, see [AG, ART]. The corresponding pocket of programs can be found in <http://www.mosek.com>.

The second difference with respect to Algorithm **(R)** is concerned with the stopping criterion. In particular (9) has to be replaced by the following condition (15). For a given pair (v, \tilde{A}) we have to check in fact the condition

$$|\operatorname{Re}(v_j^*, \tilde{A}v)| \leq 1, \quad j = 1, \dots, p. \quad (15)$$

4 Computing of the joint spectral radius: the case of nonnegative matrices **(P)**

Algorithm **(R)** can be modified for families of nonnegative matrices to improve significantly its efficiency. The corresponding Algorithm **(P)** has a very similar structure, but differs from Algorithm **(R)** in several key points. Before describing the algorithm we need to establish several auxiliary results on operators with an invariant cone.

4.1 Operators with invariant cones. Monotone extremal norms

Let K be a convex closed pointed nondegenerate cone with the apex at the origin. In the sequel we write K^* for the dual cone: $K^* = \{u \in \mathbb{R}^d \mid \inf_{x \in K} (u, x) \geq 0\}$. According to the Perron-Frobenius theorem, every operator B that leaves a cone K invariant has a positive leading eigenvalue $\lambda_{\max} = \rho(B)$ and K contains a leading eigenvector corresponding to this eigenvalue. Any leading eigenvector of B that belongs to the cone K will be referred as Perron-Frobenius eigenvector.

If all operators of the family \mathcal{M} share a common invariant cone, then Theorem 1 on the existence of invariant norms can be slightly sharpened. First, the irreducibility condition can be relaxed; second, an invariant norm can always be chosen to be monotone with respect to the invariant cone. A function g is *monotone* on a cone K if $g(x) \geq g(y)$, whenever $(x - y) \in K$. If g is a monotone norm defined on the cone K , then it is extended onto \mathbb{R}^d in a standard way: the unit ball of that norm is

$$\{x \in \mathbb{R}^d \mid \|x\| \leq 1\} = \operatorname{co}_s \{x \in K, g(x) \leq 1\}. \quad (16)$$

All extreme points of the ball defined by (16) are in the cones K and $-K$. Since the norm of any operator A is attained at an extreme point of the unit ball, we see that if A leaves K invariant, it attains its norm in the cone K . Thus,

$$\|A\| = \max_{x \in K, g(x) \leq 1} g(Ax).$$

In particular, if g is an extremal norm for a family \mathcal{M} , i.e., $\max_{i=1, \dots, m} \|A_i\| = \hat{\rho}$, then its extension defined by (16) is extremal as well. Thus, for families with a common invariant cone it suffices to construct an extremal monotone norm g on that cone.

We are going to show that there exists not only extremal, but invariant monotone norm on K . Recall, that a norm in K is invariant for \mathcal{M} if

$$\max\{\|A_1x\|, \dots, \|A_mx\|\} = \hat{\rho}\|x\|, \quad x \in K.$$

To formulate the main result we need some further notation. A hyperplane $L \subset \mathbb{R}^d$ is called a *plane of support* of a cone K if $L \cap K \neq \{0\}$ and $L \cap \text{int } K = \emptyset$. A *face* of a cone is its intersection with some plane of support. For example, a spherical cone has only one-dimensional faces (rays); the faces of the cone $K = \mathbb{R}_+^d$ are coordinate planes: $F_{i_1 \dots i_r} = \{x \in \mathbb{R}_+^d \mid x_{i_1} = \dots = x_{i_r} = 0\}, r = 1, \dots, d-1$. A face F of a cone K is *invariant* for an operator A if $AF \subset F$.

Theorem 3 *If operators of a family $\mathcal{M} = \{A_1, \dots, A_m\}$ share an invariant cone K and do not have common invariant faces of that cone, then \mathcal{M} possesses a monotone invariant norm on K .*

In the proof of Theorem 3 we use the following lemma from [P3, section 4]:

Lemma 4 [P3] *For any cone K and for any norm on this cone there is a homogeneous continuous function $\gamma(x)$ positive on the interior of K such that for every operator B leaving the cone invariant we have $\|Bx\| \geq \gamma(x)\|B\|$.*

Proof. The case $\hat{\rho} = 0$ is impossible, because in this case the operators must have a common invariant face. This fact is simple, and we omit its proof. If $\hat{\rho} > 0$, then after normalization it can be assumed that $\hat{\rho} = 1$. Let us first get an extremal norm for \mathcal{M} . Take any $e^* \in \text{int } K^*$ and for each $n \geq 1$ consider the function

$$g_n(x) = \sup_{k \geq n} \max_{A \in \mathcal{M}^k} (e^*, Ax), \quad x \in K. \quad (17)$$

Note the following properties of these functions.

1. For very n we have $g_{n+1}(x) \leq g_n(x)$, so the sequence $\{g_n\}_{n \in \mathbb{N}}$ is monotone.
2. We have

$$\sup_{x \in K, (e^*, x) = 1} g_1(x) < \infty,$$

hence, by the monotonicity of the sequence $\{g_n\}$, all g_n are uniformly bounded on the unit sphere. To prove that g_1 is bounded observe that the set $L = \{x \in K \mid g_1(x) < +\infty\}$ is either $\{0\}$, or K , or a common invariant face of K . The latter contradicts the assumption. Assume $L = \{0\}$. Then consider the compact set $S = \{x \in K \mid (e^*, x) = 1\}$. For each $j \geq 1$ we define the set

$$V_j = \left\{ x \in S \mid \max_{A \in \mathcal{M}^j} (e^*, Ax) > 2 \right\}.$$

Thus V_j consists of vectors for which some product A of length j increases the value (e^*, x) more than twice. If $L = \{0\}$, then $\cup_{j \geq 1} V_j = S$, and, since all V_j are open in S , from the compactness of S it follows that $\cup_{j=1}^N V_j = S$ for some N . This means that for every $x \in K$ there is a product A of length at most N increasing the value (e^*, x) at least twice. Applying this argument successively k times, we conclude that for every $x \in K$ there is a product Π_k of length $l_k \leq kN$ such that $(e^*, \Pi_k x) \geq 2^k (e^*, x)$, and hence $\|\Pi_k\| \geq 2^k (e^*, x) \|e^*\|^{-1}$ (the norm of Π_k is the Euclidean). Taking the power $1/l_k$ and the limit as $k \rightarrow \infty$, we obtain $\hat{\rho} \geq 2^{1/N}$, which is a contradiction. Thus, $L \neq \{0\}$, and hence $L = K$. Thus, each function g_n is bounded on S .

3. For each n the function g_n is homogeneous, positive (as a supremum of positive values) and convex on K , as a pointwise supremum of linear functionals. Thus, $g_n(\cdot)$ is a norm on K .
4. For every $x \in K$ we have $g_n(A_i x) \leq g_n(x)$, $A_i \in \mathcal{M}$. Hence for each n the norm g_n is extremal. Thus, we have established the existence of a monotone sequence of extremal norms.
5. For every operator A leaving K invariant we denote by

$$\|A\|_{e^*} = \sup_{x \in S} (e^*, Ax)$$

the operator norm corresponding to the norm $\|x\|_{e^*} = (e^*, x)$. Since $\hat{\rho} = 1$, it follows that

$$\max_{A \in \mathcal{M}^k} \|A\|_{e^*} \geq 1.$$

Now using Lemma 4, we obtain $\max_{A \in \mathcal{M}^k} \|Ax\|_{e^*} \geq \gamma(x)\|x\|_{e^*}$. This holds for each k , therefore, $g_n(x) \geq \gamma(x)\|x\|_{e^*}$ for every $x \in K$.

6. Since the sequence $\{g_n(x)\}_{n \in \mathbb{N}}$ is non-increasing and bounded below, it converges to some limit function $g(x)$. For every $x \in \text{int } K$ we have $g(x) \geq \gamma(x)\|x\|_{e^*} > 0$. Thus, the function g is convex, positively homogeneous, and invariant, i.e., possesses the property

$$g(x) = \max_{j=1, \dots, m} g(A_j x).$$

It remains to show that g is positive on K , in such case it constitutes an invariant norm. Since $g(x) > 0$ for $x \in \text{int } K$, we see that the set $L = \{x \in K \mid g(x) = 0\}$ lies on the boundary of K . This set is obviously convex, hence it is contained on a face of K . Let V be the minimal (by inclusion) face containing L . Since $A_j L \subset L$, $j = 1, \dots, m$, it follows that $A_j V \subset V$, $j = 1, \dots, m$, which contradicts the assumption. Thus, g is an invariant norm, which completes the proof. \square

Now we focus on the case of nonnegative operators, i.e., operators defined by nonnegative matrices (which means, with nonnegative entries). Each family of nonnegative operators share an invariant cone \mathbb{R}_+^d . A family of nonnegative operators is called *positively-irreducible* if they do not have common invariant faces among the coordinate planes. Applying Theorem 3 to the case $K = \mathbb{R}_+^d$, we obtain:

Corollary 1 *A family of nonnegative positively-irreducible operators possesses a monotone invariant norm on \mathbb{R}_+^d .*

Remark 6 The assumption of Corollary 1 is not restrictive, because the general case of nonnegative matrices is reduced to the case of matrices without invariant coordinate planes. If the matrices possess common invariant planes, then after a suitable permutation of the basis vectors all the matrices get a block upper-triangular form. The joint spectral radius of the matrices equals to the largest joint spectral radius of the blocks. Thus, the problem of JSR computation comes to several similar problems with nonnegative matrices of smaller dimensions. A fast polynomial procedure to realize this reduction can be found in [JPB1, section 2].

Before describing Algorithm **(P)** we formulate an analogue of Lemma 1 for nonnegative operators.

Lemma 5 *Let a nonnegative operator B have a unique simple Perron-Frobenius eigenvalue λ with an eigenvector v ; let also v^* be the Perron-Frobenius eigenvector of B^* such that $(v^*, v) = 1$. If for some nonnegative operator C one has $(v^*, Cv) > 1$, then for sufficiently large r the operator $B^r C$ has a unique simple Perron-Frobenius eigenvalue bigger than λ .*

The proof is literally the same as for Lemma 1. \square

If a cone $K \subset \mathbb{R}^d$ is fixed, then for a given set $Q \subset \mathbb{R}^d$ we denote

$$\text{co}_-(Q) = (\text{co}(Q) - K) \cap K = \{x \in K \mid x = y - z, y \in \text{co}(Q), z \in K\}.$$

If the cone K is not specified, we always assume $K = \mathbb{R}_+^d$.

Everywhere below in this section the family \mathcal{M} is assumed to be positively irreducible (see Remark 6).

4.2 Algorithm (P) versus Algorithm (R)

Algorithm (P) is similar to Algorithm (R) but has some peculiar differences which we remark in the sequel. The second has a major computational importance.

- (i) By the Perron-Frobenius theorem the candidate s.m.p. Π has a nonnegative leading eigenvalue $\lambda_{\max} = \rho(\Pi)$, and the corresponding eigenvector belongs to \mathbb{R}_+^d . We assume that $\lambda_{\max} > 0$. Hence the main assumption for Algorithm (R) holds true automatically.
- (ii) The LP problem performed in the loop at step k should be replaced by the following:

$$\left\{ \begin{array}{l} \max t_0 \\ \text{subject to } t_0 z \leq \sum_{x \in \mathcal{V}_k} t_x x \\ \text{and } \sum_{x \in \mathcal{V}_k} t_x \leq 1, \quad t_x \geq 0 \quad \forall x \in \mathcal{V}_k \end{array} \right. \quad (18)$$

where we recall that $z = \tilde{A}v$.

- (iii) In contrast to the case (R), we work now only with nonnegative vectors, and do not include the vectors $-v_i$ to the polytope P_k . We do not construct a symmetric polytope $\text{co}_s(\mathcal{V}_k)$, but a positive polytope $\text{co}_-(\mathcal{V}_k)$. So, our norm will have a unit ball $\text{co}_-(\mathcal{V}_k)$. This explains the differences between LP-problems (8) and (18).
- (iv) Condition (9) in the **Stopping criterion** should be replaced by the following:

$$(v_j^*, \tilde{A}v) \leq 1, \quad j = 1, \dots, n. \quad (19)$$

4.3 Explanations and proofs

The theoretical base of Algorithm **(P)** is actually the same as for Algorithm **(R)**. Let us only stress the distinctions. First of all, we construct a monotone extremal norm on the positive orthant \mathbb{R}_+^d , so the polytopes P_k are in the orthant, and are not centrally-symmetric. That is why we do not need to symmetrize the convex hull of \mathcal{V}_k . Consequently, LP-problem (18), in contrast to LP-problem (8), does not have extra variables $q_x, x \in \mathcal{V}_k$. The other difference is that the equality constraint $t_0 \tilde{A} v = \sum_{x \in \mathcal{V}_k} t_x x$ becomes an inequality. This means that the polytope P_k is not a convex hull of \mathcal{V}_k , but $\text{co}_-(\mathcal{V}_k)$. Thus there are two advantages of Algorithm **(P)**: 1) the number of variables and the number of constraints in the LP-problem is a half of the LP-problem in Algorithm **(R)**; 2) the polytope $\text{co}_-(\mathcal{V}_k)$ is larger than $\text{co}(\mathcal{V}_k)$, therefore this algorithm sorts out more vertices (“dead branches”) at each step, which leads to a lower complexity. In practice Algorithm **(P)** works much faster than Algorithm **(R)** (see Section 7 and Section 9).

In the worst case, if the algorithm has not terminated, one gets an approximate value of JSR from inequality

$$[\rho(\Pi)]^{1/n} \leq \hat{\rho}(\mathcal{M}) \leq (t_N)^{-1} [\rho(\Pi)]^{1/n}. \quad (20)$$

Proposition 3 *For an arbitrary positively-irreducible family \mathcal{M} we have $t_N \rightarrow 1$ as $l \rightarrow \infty$ and $N \rightarrow \infty$ in estimate (20).*

The proof is the same as for Proposition 1 with the use of Theorem 3 instead of Theorem 1. The proof of the efficiency of the stopping criterion (Proposition 4) is also the same as for Proposition 2:

Proposition 4 *Assume the Perron-Frobenius eigenvalue of Π is unique and simple. If the assumption of the algorithm is wrong (i.e., Π is not an s.m.p.) then for every $j = 1, \dots, n$ condition (19) is violated at some step. Conversely, if condition (19) is violated at some step for some j , then Π is not an s.m.p.*

Remark 7 In Algorithm **(P)** the family \mathcal{M} is assumed to be positively irreducible. Actually, this was done for the sake of simplicity. The algorithm can be applied to arbitrary nonnegative families. If the algorithm terminates after k th iteration, and the set \mathcal{V}_k does not lie in a coordinate plane of a smaller dimension, then P_{k-1} is an extremal polytope, and Π is an s.m.p. That condition means that for each $i = 1, \dots, d$ there is a vector from \mathcal{V}_k with strictly positive i th coordinate. This simple condition allows us to apply Algorithm **(R)** to arbitrary family, without preliminary checking its positive irreducibility. Nevertheless, if the family \mathcal{M} is reducible, then it is always advisable to factorize it before starting the algorithm, because this significantly reduces the dimension (see Remark 6). Especially as the factorization is realized by a fast polynomial routine [JPB1].

5 The criterion for finite termination of Algorithms **(R)**, **(C)** and **(P)**.

Algorithms **(R)**, **(C)**, and **(P)** compute JSR by step-by-step constructing a polytope norm. If the algorithm terminates within finite time, then it produces an

extremal polytope, and, hence, proves that the chosen product Π is an s.m.p. If it does not terminate, then it gives upper and lower bounds for JSR that converge to the exact value. An important issue is the following: what are the conditions for the family \mathcal{M} and for the product Π , under which the algorithm terminates and produces the extremal polytope? Conditions guaranteeing the convergence of the algorithm to an extremal polytope norm have been discussed in [GWZ, GZ2]. We give here a further result which is related to those obtained in the mentioned papers.

Certainly, the product Π must be spectral maximizing for that. This condition, however, does not guarantee the convergence of the algorithm. It appears that a bit stronger condition solves the problem completely: it is both sufficient and necessary. The product Π has to be not just maximizing but *dominant*. To formulate the criterion we need some further notation.

Let $\mathcal{M} = \{A_1, \dots, A_m\}$ be a given family of operators, $\Pi = A_{d_n} \cdots A_{d_1}$ be some product, which is not a power of a shorter product, $n \geq 1$. We denote $\tilde{A}_i = [\rho(\Pi)]^{-1/n} A_i$, $\tilde{\mathcal{M}} = \{\tilde{A}_1, \dots, \tilde{A}_m\}$ and $\tilde{\Pi} = \tilde{A}_{d_n} \cdots \tilde{A}_{d_1}$. Clearly, the spectral radius of any power of $\tilde{\Pi}$ or of any power of its cyclic permutation is 1.

Definition 3 A product $\Pi \in \mathcal{M}^n$ is called *dominant* for the family \mathcal{M} if there is $q < 1$ such that the spectral radius of every product of operators of the normalized family $\tilde{\mathcal{M}}$, that is not a power of $\tilde{\Pi}$ nor a power of its cyclic permutations, is smaller than q .

Obviously, the dominant product along with all its cyclic permutations are all s.m.p., but vice versa.

The following theorem gives a sharp criterion on the family \mathcal{M} ensuring that our algorithm produces an extremal polytope.

Theorem 4 For each of Algorithms (R), (C) and (P) the following holds:

the algorithm terminates within finitely many iterations if and only if Π is dominant for \mathcal{M} , and its leading eigenvalue is unique and simple.

The proof is in Appendix.

Corollary 2 If a family \mathcal{M} possesses a dominant product, whose leading eigenvalue is unique and simple, then it has an extremal polytope.

Remark 8 Theorem 4 remains true even if we do not apply the stopping criterion in the algorithms.

Remark 9 Theorem 4 implies that if the algorithm terminates within finite time, then the family \mathcal{M} possesses a dominant product. Actually the algorithm ensures that a chosen product Π is dominant. In numerical examples from applications (Sections 8) and from randomly generated matrices (Section 9) most of matrix families possess dominant products.

The assumption on a dominant product allows to exclude a limit spectrum maximizing product, that is a matrix in the closure of the multiplicative semigroup of \mathcal{M} with spectral radius equal to 1 (see [GZ4]).

Consider in fact the following example. Let $\mathcal{M} = \{A_1, A_2\}$:

$$A_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad A_2 = \frac{4}{5} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

We can prove that $\hat{\rho}(\mathcal{M}) = 1 + \frac{1}{\sqrt{5}}$ and there is a unique *finite* spectrum maximizing product $\Pi = A_1 A_2$ such that $\rho(\tilde{P}) = 1$ (apart from its cyclic permutation $A_2 A_1$ and their powers). Nevertheless the product is not dominant. In fact the sequence of products $\tilde{Q}_k = \tilde{A}_1 \left(\tilde{A}_1 \tilde{A}_2 \right)^k$ is convergent and such that

$$\lim_{k \rightarrow \infty} \tilde{Q}_k = \begin{pmatrix} \frac{\sqrt{5}+1}{4} & \frac{1}{2} \\ \frac{\sqrt{5}-1}{4} & \frac{3-\sqrt{5}}{4} \end{pmatrix} := \tilde{Q}_\infty$$

which is such that $\rho(\tilde{Q}_\infty) = 1$.

This implies that P is not dominant; the matrix \tilde{Q}_∞ is indeed a limit spectrum maximizing product of the normalized family $\tilde{\mathcal{M}}$.

Indeed the algorithms **(R)** and **(P)** do not converge when applied to this example. However, if we modify the algorithms and remove a vector when it lies on the boundary of the polytope P_{k-1} , then we obtain a finite convergence also in this case.

6 Computing the lower spectral radius. Algorithm (L).

In this section we describe a method for the exact computation of the lower spectral radius of a finite family of matrices.

6.1 Antinorms on convex cones

To extend our approach to computing the lower spectral radius, first of all we need the notion of extremal norm for this case. One can define it by the inequality $\min_{A_i \in \mathcal{M}} \|A_i x\| \geq \tilde{\rho} \|x\|$, $x \in \mathbb{R}^d$. However, simple examples show that such a norm may not exist even for very “good” families \mathcal{M} (for instance, irreducible families of positive matrices). The reason is that the function $x \mapsto \min_{A_i \in \mathcal{M}} \|A_i x\|$ may not be convex, in which case it is not a norm (in contrast to the situation with JSR, when the function $x \mapsto \max_{A_i \in \mathcal{M}} \|A_i x\|$ is always a norm). One of the ways to generalize the notion of extremal norm for the lower spectral radius is to consider concave positive homogeneous functionals on \mathbb{R}^d instead of convex ones (i.e., instead of norms). However, such functionals do not exist. Indeed, if f is concave, then $f(x) + f(-x) \leq f(0) = 0$ (homogeneity failure), hence f cannot be positive. Nevertheless, extremal concave “norms” can be defined, provided all operators of the family \mathcal{M} share an invariant cone. In particular, this can be done for families of nonnegative matrices. As in the previous section, K is a convex closed pointed nondegenerate cone with an apex at the origin.

Definition 4 *An antinorm is a continuous nonnegative nontrivial (not identical zero) concave positively-homogeneous function on a cone K .*

From the concavity it easily follows that an antinorm can vanish only on the boundary of K . An antinorm f is called *positive* if $f(x) > 0$ for all $x \in K \setminus \{0\}$. So, f is positive, whenever it is positive on the boundary. This is well known that a concave function is continuous at each interior point of its domain. Hence

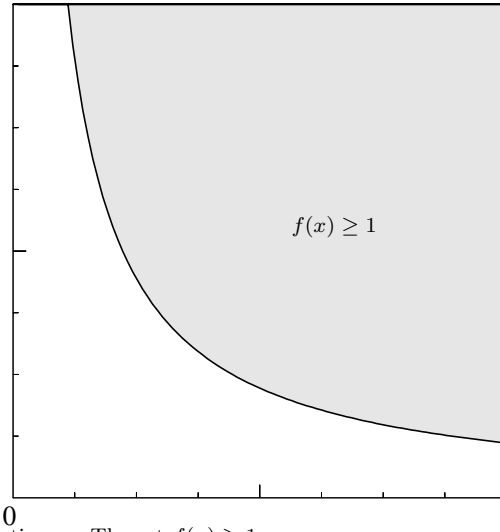


Fig. 1 Example of antinorm. The set $f(x) \geq 1$.

the continuity condition for antinorms can be relaxed to the continuity on the boundary.

Let us observe some basic properties of antinorms. First of all, every antinorm is asymptotically bounded above by every norm in \mathbb{R}^d :

Lemma 6 *For any antinorm f and for any norm $\|\cdot\|$ there is a constant C such that*

$$f(x) \leq C \|x\|, \quad x \in K.$$

Proof. Since f is continuous, the value $C = \sup_{x \in K, \|x\|=1} f(x)$ is finite. Now by the homogeneity the lemma follows. \square

Consider now a family of operators $\mathcal{M} = \{A_1, \dots, A_m\}$ that share an invariant cone K .

Proposition 5 *If for some antinorm f and for a constant λ we have $f(A_i x) \geq \lambda f(x)$, $x \in K$, $A_i \in \mathcal{M}$, then $\check{\rho} \geq \lambda$.*

Proof. Applying Proposition 5 for an arbitrary point $e \in \text{int } K$, $\|e\| = 1$, we get $\|A_{d_k} \cdots A_{d_1} e\| \geq C^{-1} f(A_{d_k} \cdots A_{d_1} e) \geq C^{-1} \lambda^k f(e)$. Thus,

$$\min_{d_k, \dots, d_1} \|A_{d_k} \cdots A_{d_1}\| \geq C^{-1} f(e) \lambda^k.$$

Taking the power $1/k$ and the limit as $k \rightarrow \infty$, we conclude the proof. \square

Definition 5 *An antinorm is called extremal if $f(A_i x) \geq \check{\rho} f(x)$, $x \in K$, $A_i \in \mathcal{M}$.*

Similar to monotone norms, an antinorm is called *monotone* if $f(x) \geq f(y)$, whenever $x - y \in K$.

Theorem 5 *For every family of matrices with a common invariant cone K there exists a monotone extremal antinorm on K .*

Proof. If $\check{\rho} = 0$, then any antinorm suffices. If $\check{\rho} > 0$, then after normalization it can be assumed that $\check{\rho} = 1$. Take any $e^* \in \text{int } K^*$ and consider the function

$$f(x) = \inf_{k \geq 0} \min_{A \in \mathcal{M}^k} (e^*, Ax), \quad x \in K. \quad (21)$$

This function is concave, homogeneous and monotone, and $f(A_i x) \geq f(x)$ for each $A_i \in \mathcal{M}$. It remains to show that f is not an identical zero. Consider the compact set $S = \{x \in K \mid (e^*, x) = 1\}$ and for each $k \geq 1$ define the set

$$U_k = \left\{ x \in S \mid \min_{A \in \mathcal{M}^k} (e^*, Ax) < \frac{1}{2} \right\}.$$

If $f \equiv 0$, then $\bigcup_{k \geq 1} U_k = S$, and, since all U_k are open in S , from the compactness of S it follows that $\bigcup_{k=1}^N U_k = S$ for some N . This means that for every $x \in K$ there is a product A of length at most N reducing the value (e^*, x) at least twice. Applying this argument k times, we obtain that for every $x \in K$ there is a product Π_k of length $l_k \leq kN$ such that $(e^*, \Pi_k x) \leq 2^{-k} (e^*, x)$. Note that if $x \in \text{int } K$, then there is a constant C that depends on e^* and on x , and such that $\|B\| \leq C(e^*, Bx)$ for any operator B that leaves K invariant (see, for instance [P3]). Thus, $\|\Pi_k\| \leq C 2^{-k} (e^*, x)$ for each k . Taking the power $1/l_k$ and the limit as $k \rightarrow \infty$, we get $\check{\rho} \leq 2^{-1/N}$, which is a contradiction. \square

Applying this theorem for the case $K = \mathbb{R}_+^d$, we obtain:

Corollary 3 *For an arbitrary family of nonnegative matrices there is a monotone extremal antinorm on \mathbb{R}_+^d .*

In the algorithm we need the following analogue of Lemma 5, whose proof is the same.

Lemma 7 *Under the assumptions of Lemma 5, if for some nonnegative operator C one has $(v^*, Cv) < 1$ then for sufficiently large n the operator $B^n C$ has a unique simple Perron-Frobenius eigenvalue smaller than λ .*

Now we are ready to describe Algorithm **(L)** of LSR computation for nonnegative matrices. We use the notation $\text{co}_+(X) = \text{co}(X) + K = \{x + h \mid x \in \text{co}(X), h \in K\}$, where X is a subset of \mathbb{R}^d and $K \subset \mathbb{R}^d$ is a cone. If X is finite, then $\text{co}_+(X)$ will be referred as an *infinite polytope*. Thus, an infinite polytope is a set $P + K$, where K is a cone and P is a polytope. In the algorithm we always assume $K = \mathbb{R}_+^d$.

A product $\Pi \in \mathcal{M}^n$ is called the *spectral lowest product (s.l.p.)* if $[\rho(\Pi)]^{1/n} = \check{\rho}(\mathcal{M})$. We shall also call it a *spectrum minimizing product*, but always use the abbreviation s.l.p. to avoid confusion with the spectral maximizing product (s.m.p.). By inequality (4) a product Π is an s.l.p. iff $[\rho(\Pi)]^{1/n} \leq \check{\rho}(\mathcal{M})$. Proposition 5 implies that if there is an antinorm $f : \mathbb{R}_+^d \rightarrow \mathbb{R}_+$ such that $f(A_j x) \geq [\rho(\Pi)]^{1/n} f(x)$, $x \in \mathbb{R}_+^d$, then $[\rho(\Pi)]^{1/n} = \check{\rho}(\mathcal{M})$, and f is extremal. The main idea of the algorithm is to select a candidate Π for s.l.p. (by a reasonable exhaustion) and then to prove that it is actually an s.l.p. The proof is by step-by-step constructing an extremal infinite polytope, which generates an extremal antinorm.

6.2 Algorithm (L)

Initialization. We have an arbitrary family $\mathcal{M} = \{A_1, \dots, A_m\}$. For some (as large as possible) l we look over all products Π of length $\leq l$ and take one with the smallest value $[\rho(\Pi)]^{1/n}$, where n is the length of the product. We take the shortest product possessing this property and denote it as $\Pi = A_{d_n} \cdots A_{d_1}$. If $\rho(\Pi) = 0$, then $\hat{\rho}(\mathcal{M}) = 0$, and the algorithm terminates. So, we assume $\rho(\Pi) > 0$. By the Perron-Frobenius theorem Π has a positive leading eigenvalue, and the corresponding eigenvector belongs to \mathbb{R}_+^d . We normalize the family \mathcal{M} as $\tilde{A}_i = [\rho(\Pi_1)]^{-1/n} A_i$, $\tilde{M} = \{\tilde{A}_i\}_{i=1}^m$. The leading eigenvalue of the operator $\tilde{\Pi} = \tilde{A}_{d_n} \cdots \tilde{A}_{d_1}$ equals to 1.

Let $\tilde{\Pi}_1 = \tilde{\Pi}$, $\tilde{\Pi}_i = \tilde{A}_{d_{i-1}} \cdots \tilde{A}_{d_1} \tilde{A}_{d_n} \cdots \tilde{A}_{d_i}$ be a cyclic permutation of $\tilde{\Pi}_1$, $i = 2, \dots, n$. We take a Perron-Frobenius eigenvector v_1 of $\tilde{\Pi}_1$ (if it is not unique, take any of them), and

$$v_i = \tilde{A}_{d_{i-1}} \cdots \tilde{A}_{d_1} v_1.$$

Thus, v_i is a Perron-Frobenius eigenvector of $\tilde{\Pi}_i$ with the eigenvalue 1.

In case the leading eigenvalue $\lambda_{\max} = 1$ is unique and simple, we also need a dual system of vectors: v_1^* the leading eigenvector of $\tilde{\Pi}_1^*$ normalized as $(v_1^*, v_1) = 1$ (Remark 1), and

$$v_i^* = \tilde{A}_{d_1}^* \cdots \tilde{A}_{d_{i-1}}^* v_1^*, \quad i = 2, \dots, n.$$

Thus, v_i and v_i^* are the Perron-Frobenius eigenvectors of Π_i and Π_i^* respectively, and $(v_i^*, v_i) = 1$.

Set $k = 0$. We set $\mathcal{V}_0 = \mathcal{U}_0 = \{v_1, \dots, v_n\}$ and $\mathcal{R}_0 = \{(v_i, \tilde{A}_p) \mid i = 1, \dots, n; p = 1, \dots, m, p \neq d_i\}$.

Main loop

For $k \geq 1$. We have finite sets $\mathcal{V}_{k-1} \subset \mathbb{R}^d$, $\mathcal{U}_{k-1} \subset \mathcal{V}_{k-1}$, and $\mathcal{R}_{k-1} \subset \mathcal{U}_{k-1} \times \tilde{\mathcal{M}}$. Put $\mathcal{V}_k = \mathcal{V}_{k-1}$ and $\mathcal{U}_k = \emptyset$.

We successively take all pairs $(v, \tilde{A}) \in \mathcal{R}_{k-1}$. If for a given pair $\tilde{A}v = 0$, then we stop the algorithm, it is inapplicable for this case. If $z = \tilde{A}v \neq 0$, then we solve the following LP problem with variables t_0 and $\{t_x\}_{x \in \mathcal{V}_k}$:

$$\begin{cases} \min t_0 \\ \text{subject to } t_0 z \geq \sum_{x \in \mathcal{V}_k} t_x x \\ \text{and } \sum_{x \in \mathcal{V}_k} t_x \geq 1, \quad t_x \geq 0 \quad \forall x \in \mathcal{V}_k. \end{cases} \quad (22)$$

The value of the problem, i.e., $\min t_0$ will be denoted by $t_{\{v, \tilde{A}\}}$. Thus, for every pair $(v, \tilde{A}) \in \mathcal{R}_{k-1}$ we have a nonnegative number $t_{\{v, \tilde{A}\}}$, which may take value $+\infty$, when the system of inequality constraints has no solution.

If $t_{\{v, \tilde{A}\}} < 1$, then leave the sets \mathcal{V}_k and \mathcal{U}_k as they are, take the next pair $(v, \tilde{A}) \in \mathcal{R}_{k-1}$ and consider problem (22) for it.

Otherwise **If** $1 \leq t_{\{v, \tilde{A}\}}$, then

If the leading eigenvalue of Π is unique and simple, we apply the following stopping criterion:

Stopping criterion

We check the condition

$$(v_j^*, \tilde{A}v) \geq 1, \quad j = 1, \dots, n. \quad (23)$$

If (23) is satisfied, then we set $\mathcal{V}_k = \mathcal{V}_k \cup \{\tilde{A}v\}$, $\mathcal{U}_k = \mathcal{U}_k \cup \{\tilde{A}v\}$, take the next pair $(v, \tilde{A}) \in \mathcal{R}_{k-1}$ and consider problem (22) for it.

Otherwise If (23) is not satisfied, then our assumption is wrong, Π is not an s.l.p., and $\check{\rho}(\tilde{M}) < 1$ (Lemma 7). We stop the algorithm and go either to the **Final step**, or back to the **Initialization**. In the latter case we need to find another pretender to s.l.p. The first opportunity is to increase l and to look over all products of a bigger length. Lemma 7 provides also a different approach. We take an index j , for which $(v_j^*, \tilde{A}v) < 1$. Applying Lemma 7, we conclude that there is r such that $\lambda_{\max}(\tilde{\Pi}_j^r \tilde{A}_{s_q} \cdots \tilde{A}_{s_1}) < 1$, where $\tilde{A}_{s_q} \cdots \tilde{A}_{s_1} v_j = \tilde{A}v$. We take the new initial product $\Pi = \Pi_j^r \tilde{A}_{s_q} \cdots \tilde{A}_{s_1}$ and restart the algorithm.

End If

Otherwise If the leading eigenvalue of Π is not unique or multiple, then we do not apply the stopping criterion, and set $\mathcal{V}_k = \mathcal{V}_k \cup \{\tilde{A}v\}$, $\mathcal{U}_k = \mathcal{U}_k \cup \{\tilde{A}v\}$, take the next pair $(v, \tilde{A}) \in \mathcal{R}_{k-1}$ and consider problem (22) for it.

End If

The k th step is over, when all pairs $(v, \tilde{A}) \in \mathcal{R}_{k-1}$ are exhausted.

If $\mathcal{U}_k = \emptyset$, then $\check{\rho}(\tilde{M}) = 1$, and so $\check{\rho}(\mathcal{M}) = [\rho(\Pi)]^{1/n}$. The extremal infinite polytope is $P_{k-1} = \text{co}_+(\mathcal{V}_k)$, and the product Π is an s.l.p. for \mathcal{M} . The algorithm terminates after the k th step.

Otherwise If $\mathcal{U}_k \neq \emptyset$, then we set $\mathcal{R}_k = \mathcal{U}_k \times \tilde{\mathcal{M}}$ and go to the $(k+1)$ st step.

End If

End For

Final step. If the algorithm has not terminated, then we stop it after some N steps, denote $t_N = \max_{(v, \tilde{A}) \in \mathcal{R}_{N-1}} t_{\{v, \tilde{A}\}}$, and have the following estimate for the lower spectral radius:

$$(t_N)^{-1} [\rho(\Pi)]^{1/n} \leq \check{\rho}(\mathcal{M}) \leq [\rho(\Pi)]^{1/n}. \quad (24)$$

End of Algorithm (L)

6.3 Explanations and proofs

The algorithm produces a sequence of embedded infinite polytopes $P_1 \subset P_2 \subset \dots$ such that $P_{j+1} = \text{co}_+\{\tilde{A}_1 P_j, \dots, \tilde{A}_m P_j\}$ and $P_j \subset P_{j+1}$ for every j . If the algorithm terminates after the k th step, then $P_k = P_{k-1}$. The k th step is

actually needed only to ensure that the infinite polytope P_{k-1} is extremal, i.e., $\tilde{A}_j P_{k-1} \subset P_{k-1}$, $j = 1, \dots, m$. Since $0 \notin P_{k-1}$ (otherwise at some step we had $\tilde{A}v = 0$, in which case the algorithm would be stopped by the end of the k th step), the antinorm $f_{k-1}(x) = \sup \{t^{-1} \mid tx \in P_{k-1}\}$ is well-defined on \mathbb{R}^d . Since $f_{k-1}(\tilde{A}_j x) \geq f_{k-1}(x)$ for every x , we see that $\check{\rho}(\tilde{\mathcal{M}}) \geq 1$. On the other hand, $\check{\rho}(\tilde{\mathcal{M}}) \leq [\rho(\tilde{\Pi})]^{1/n} = 1$. Thus $\check{\rho}(\tilde{\mathcal{M}}) = 1$ and so $\check{\rho}(\mathcal{M}) = [\rho(\Pi)]^{1/n}$. Thus, if the algorithm terminates within finite time, then the s.l.p. and the exact value of LSR are found. In this case P_{k-1} is an extremal infinite polytope and f_{k-1} is an extremal antinorm.

Although the extremal antinorm is obtained numerically, the results are actually exact, because the algorithm removes only those points v , for which the strict inequality $f_i(v) > 1$ holds, where f_i is the antinorm generated by the current polytope $P_i = \text{co}_+(\mathcal{V}_i)$.

If the algorithm does not terminate within finitely many steps, then we have estimate (24) to get an approximate value of the LSR. The right hand side inequality is obvious, the left hand side is equal to $\min_{j=1, \dots, m} \inf_{f_{N-1}(x)=1} f_{N-1}(A_j x) \leq \check{\rho}(\mathcal{M})$, from which the estimate follows.

6.4 Efficiency results for Algorithm (L)

Let us start with the stopping criterion. If the stopping criterion is applicable (i.e., the leading eigenvalue of Π is unique and simple), then it always determines, whether Π is an s.l.p. or not.

Proposition 6 *Assume the Perron-Frobenius eigenvalue of Π is unique and simple. If the assumption of the algorithm is wrong (i.e., Π is not an s.l.p.) then for every $j = 1, \dots, n$ condition (23) is violated at some step. Conversely, if condition (23) is violated at some step for some j , then Π is not an s.l.p.*

Proof. The sufficiency follows from Lemma 7. To show the necessity, we assume that $\check{\rho}(\tilde{\mathcal{M}}) < 1$. Then there is a product $C \in \tilde{\mathcal{M}}^s$ such that $\rho(C) < 1$. This yields $C^r \rightarrow 0$, and hence $C^r v_1 \rightarrow 0$ as $r \rightarrow \infty$. Consequently, for every j one has $(v_j^*, C^r v_1) < 1$, whenever r is large enough. Since the point $C^r v_1$ belongs to the infinite polytope P_{rs} , we see that $\inf_{x \in P_{rs}} (v_j^*, x) < 1$. This infimum is attained at some vertex v of P_{rs} , hence that vertex violates condition (23). \square

Now let us analyze estimate (24). In contrast to the algorithms for JSR computation, the lower bound $(t_N)^{-1}[\rho(\Pi)]^{1/n}$ may not converge to $\check{\rho}$ at all, even if the family \mathcal{M} is positively irreducible. There are simple examples already in the dimension $d = 2$. The reason is that some product $A \in \mathcal{M}^l$ may have the leading eigenvector v on the boundary of the invariant cone \mathbb{R}_+^d i.e., have some zero entries. If the corresponding leading eigenvalue is unique and simple, then for each $j = 1, \dots, n$ the sequence $A^r v_j$ converges to $t_j v$ as $r \rightarrow \infty$, where $t_j \geq 0$ depends only on j . This means that some vertices of P_k approach closer and closer to the boundary as $k \rightarrow \infty$. In this case Algorithm (L) is useless: it gives neither an extremal infinite polytope nor good lower bound for $\check{\rho}$. We suggest two methods

to avoid this situation. The first one is to impose a *second invariant cone* assumption. Second, is to enlarge the invariant cone \mathbb{R}_+^d by adding *extra directions*. Let us begin with the first one.

6.5 Convergence of the algorithm. Case 1; the second invariant cone.

A cone \tilde{K} is *embedded* to a cone K if $(\tilde{K} \setminus \{0\}) \subset \text{int } K$. The pair (\tilde{K}, K) will be refereed as an *embedded pair*.

Definition 6 *An embedded pair is invariant for a family \mathcal{M} if both its cones are invariant for this family.*

An embedded invariant pair of cones for a given family will be called an *invariant pair*. For a family \mathcal{M} of nonnegative operators (i.e., operators defined by nonnegative matrices) we say that \tilde{K} is a *second invariant cone* if $(\tilde{K}, \mathbb{R}_+^d)$ is an invariant pair. So, the cone \tilde{K} is embedded in \mathbb{R}_+^d and invariant for \mathcal{M} . A simple sufficient condition for the existence of a second invariant cone is the so-called *eventual positivity* of a matrix family.

Definition 7 *A nonnegative family \mathcal{M} is called eventually positive if there is k such that all matrices of the family \mathcal{M}^r are positive for all $r \geq k$.*

In particular, if all matrices of \mathcal{M} are positive, then \mathcal{M} is eventually positive. The following trivial fact clarifies the notion of eventual positivity:

Lemma 8 *A family \mathcal{M} is eventually positive iff its matrices have neither zero columns nor zero rows, and there is k such that all matrices of \mathcal{M}^k are positive.*

Lemma 9 *Every eventually positive family possesses a second invariant cone.*

Proof. A conic hull of the set $\cup_{A \in \mathcal{M}^k} AK$, where $K = \mathbb{R}_+^d$ is the second invariant cone for \mathcal{M} . □

The key property of embedded pairs is formulated in the following lemma.

Lemma 10 *If (\tilde{K}, K) is an embedded pair, then every antinorm on K is continuous and strictly positive on \tilde{K} .*

Proof. Any nonnegative concave function, which is not an identical zero, is continuous and positive at any internal point of its domain. □

Now we can prove the existence of *invariant antinorms* in the interior cone.

Definition 8 *Let a family \mathcal{M} has a common invariant cone K . An antinorm $f : K \rightarrow \mathbb{R}_+$ is called invariant, if there is a constant $\lambda \geq 0$ such that*

$$\min_{j=1, \dots, m} f(A_j x) = \lambda f(x), \quad x \in K.$$

Theorem 6 *If the family \mathcal{M} possesses an embedded pair (\tilde{K}, K) , then it has a positive monotone invariant antinorm on \tilde{K} . For any invariant antinorm on \tilde{K} we have $\lambda = \check{\rho}(\mathcal{M})$.*

Let us recall that for extremal antinorms we have $\min_{j=1,\dots,m} f(A_j x) \geq \check{\rho} x$. It becomes an equality if the antinorm is invariant.

By Theorem 5 an extremal antinorm always exists, whenever the operators share an invariant cone. For the invariant antinorm this is not the case. There are simple examples of irreducible pairs of nonnegative 2×2 -matrices that do not have an invariant antinorm. So, the embedded pair assumption is essential in Theorem 6. In the proof of Theorem 6 we use the following simple fact from [P3, section 4].

Lemma 11 *For every pair of embedded cones (\tilde{K}, K) and for every norm in \mathbb{R}^d there is a constant γ such that for any operator B with these invariant cones and for each $x \in \tilde{K}$ one has $\|Bx\| \geq \gamma \|B\| \|x\|$.*

Proof of Theorem 6. Let f be an invariant antinorm. For each k we have $\min_{A \in \mathcal{M}^k} f(Ax) = \lambda^k f(x)$. On the other hand, combining Lemmas 6 and 11 we see that for any operator B that preserves the cones K and \tilde{K} one has

$$C_0 \|B\| \|x\| \geq f(Bx) \geq C_0^{-1} \|B\| \|x\|, \quad x \in \tilde{K},$$

where C_0 does neither depend on B nor on x .

A nontrivial concave nonnegative function is strictly positive on the interior of its domain. Therefore, $f(x) > 0$ for all $x \in \tilde{K} \setminus \{0\}$. By the compactness argument it follows that $f(z) \geq C_1 \|z\|$ for any $z \in \tilde{K}$, where $C_1 > 0$ does not depend on z . Hence, applying Lemma 11, we conclude that $f(Bx) \geq C_1 \|Bx\| \geq C_1 \gamma \|B\| \|x\|$.

Therefore $\min_{A \in \mathcal{M}^k} \|Ax\| \asymp \lambda^k$, and therefore $\check{\rho} = \lambda$. Now let us prove the existence of an invariant antinorm. It suffices to consider the case $\check{\rho}(\mathcal{M}) = 1$. By Theorem 5 there is a monotone extremal antinorm f_0 , for which $\min_{A \in \mathcal{M}} f_0(Ax) \geq f_0(x)$, $x \in \tilde{K}$. Let $f_j(x) = \min_{A \in \mathcal{M}} f_{j-1}(Ax)$, $j \in \mathbb{N}$. This is a nondecreasing sequence of antinorms. If for some $x \in \tilde{K}$ we have $f_j(x) \rightarrow +\infty$ as $j \rightarrow \infty$, then this holds for all nonzero $x \in \tilde{K}$, and hence $\check{\rho}(\mathcal{M}) > 1$. Thus, the sequence $\{f_j\}_{j \in \mathbb{N}}$ is bounded, hence it converges pointwise to some function f , which is a monotone invariant antinorm. \square

Applying now invariant antinorms we can prove the convergence results for Algorithm (L). We start with inequality (24).

Proposition 7 *If the family \mathcal{M} possesses a second invariant cone $\tilde{K} \subset \mathbb{R}_+^d$, and $v_1 \in \tilde{K}$, then for estimate (24) we have $t_N \rightarrow 1$ as $l \rightarrow \infty$ and $N \rightarrow \infty$.*

Thus, if \mathcal{M} has a second invariant cone, then Algorithm (L) is always applicable for, at least, approximate computation of LSR. It either finds the value of LSR or provides lower and upper bounds for it; those bounds are arbitrarily close to each other, whenever both l and N are large enough.

In the proof we use Dini's theorem (see [Ru, theorem 7.13]) and the following analogue of the Minkowski norm for concave functionals. We call a convex closed set $D \subset \mathbb{R}_+^d$ *admissible* if it does not contain the origin, and if with every point $x \in D$ it contains all points $y \geq x$. In particular, all infinite polytopes not containing the origin are admissible. The *Minkowski antinorm* associated to an admissible set $D \subset \mathbb{R}_+^d$ is defined as $f_D(x) = \sup \{t^{-1} \mid t > 0, tx \in D\}$, $x \in \mathbb{R}_+^d$. For any admissible set D the function f_D is a positive monotone antinorm on \mathbb{R}_+^d .

Proof of Proposition 7. Assume first that $\check{\rho}(\tilde{\mathcal{M}}) = 1$, i.e., that Π is an s.l.p. The algorithm produces the infinite polytopes $\{P_k\}_{k \in \mathbb{N}}$ such that $P_k \subset P_{k+1} = \text{co}_+ \{\tilde{A}_1 P_k, \dots, \tilde{A}_m P_k\}$. All their vertices are in \tilde{K} , because $v_1 \in \tilde{K}$. By Theorem 6 there is a positive invariant antinorm f_0 on \tilde{K} , for which

$$f_0(x) = \min_{j=1, \dots, m} f_0(\tilde{A}_j x), \quad x \in \tilde{K}.$$

Therefore, for all vertices v of the polytopes P_k one has $f_0(v) \geq f_0(v_1)$. Since f_0 is positive on \tilde{K} , by the compactness argument it follows that $f_0(x) \geq C\|x\|$, $x \in \tilde{K}$, where $C > 0$ is a constant. Whence, all polytopes P_k are uniformly separated from zero: they do not intersect the ball of radius $C\|v_1\|$ centered at the origin. Consequently, the sequence $\{f_k\}_{k \in \mathbb{N}}$ of Minkowski antinorms generated by the infinite polytopes $\{P_k\}_{k \in \mathbb{N}}$ is non-decreasing and bounded. So, it converges pointwise to a positive monotone antinorm f . By Dini's theorem [Ru, theorem 7.13], this convergence is uniform on the set $S = \{x \in \tilde{K} \mid f(x) = 1\}$. Thus, $f_k(x) \rightarrow 1$ uniformly for $x \in S$, as $k \rightarrow \infty$. Hence, there is N_ε such that $f_{N-1}(x) \geq 1 - \varepsilon$ for all $x \in S$, whenever $N \geq N_\varepsilon$. Hence, $(t_N)^{-1} = \inf_{x \in P_N} f_{N-1}(x) \geq \inf_{x \in S} f_{N-1}(x) \geq 1 - \varepsilon$, which completes the proof for the case $\check{\rho}(\tilde{\mathcal{M}}) = 1$. The transfer to the general case is realized in the same way as in the proof of Proposition 1. \square

Corollary 4 *Each of the following conditions is sufficient for the convergence $t_N \rightarrow 1$ as $l \rightarrow \infty$ and $N \rightarrow \infty$:*

- 1) *the family \mathcal{M} is eventually positive;*
- 2) *the family \mathcal{M} has a second invariant cone, and the leading eigenvalue of Π is simple.*

Proof. Observe that if a family is eventually positive, then every product Π of its matrices has a unique simple largest by modulo eigenvalue. This eigenvalue is positive, and the corresponding eigenvector is strictly positive. To see this note that the matrix Π is obviously primitive (i.e., it is nonnegative and some power Π^k is strictly positive). A primitive matrix always has a unique simple largest by modulo eigenvalue, which is positive, and the corresponding eigenvector is strictly positive [HJ, chapter 8]. If \mathcal{M} is eventually positive, then for each $A \in \mathcal{M}^k$ and every $v \in \mathbb{R}_+^d$ the vector Av belongs to the interior cone \tilde{K} , which is a conic hull of the set $\bigcup_{d_k, \dots, d_1} A_{d_k} \cdots A_{d_1}(\mathbb{R}_+^d)$. Hence, \tilde{K} contains the leading eigenvector of any product Π of matrices from \mathcal{M} .

If \mathcal{M} has a second invariant cone \tilde{K} , then, by the Perron-Frobenius theorem, \tilde{K} contains some of the leading eigenvectors of Π . If the leading eigenvalue is simple, then $v_1 \in \tilde{K}$. \square

6.6 The criterion for finite termination of Algorithm (L).

Now we are ready to prove a sharp criterion ensuring that Algorithm (L) produces an extremal infinite polytope. It looks similar to Theorem 4 and use the notion of under-dominant product.

Definition 9 A product $\Pi \in \mathcal{M}^n$ is called *under-dominant* for the family \mathcal{M} if there is $p > 1$ such that the spectral radius of every product of operators of \mathcal{M} , that is not a power of Π nor a power of its cyclic permutations, is bigger than p .

Theorem 7 Assume the family \mathcal{M} is eventually positive. Algorithm (L) terminates within finitely many iterations if and only if Π is under-dominant for \mathcal{M} .

The proof of Theorem 7 is in Appendix 10.

Remark 10 If the family \mathcal{M} is eventually positive, then every product $\Pi \in \mathcal{M}^n$ is a *primitive matrix*, i.e., some of its powers is positive.

This is well known that the leading eigenvalue of a primitive matrix is always unique and simple (see, for instance, [HJ, chapter 8]). That is why in Theorem 7 we do not need the uniqueness and simplicity of the leading eigenvalue assumption, in contrast to Theorem 4.

Remark 11 Let us stress again that Algorithm (L) can produce extremal polytopes for nonnegative families that have no second invariant cone or not eventually positive. We will see some examples in Section 7 and 8. The only difference is that we have not succeeded in finding a reasonable criterion for that case. As for Theorem 7, the eventual positivity assumption is essential and cannot be omitted.

Corollary 5 If an eventually positive family \mathcal{M} possesses an under-dominant product, then it has an extremal infinite polytope.

6.7 Case 2. Modification of Algorithm (L)

In the previous subsection we showed that if the family \mathcal{M} has a second invariant cone (in particular, if this family is eventually positive), then the algorithm is always applicable. It may converge within finite time, in which case it produces the extremal infinite polytope and finds the exact value of LSR. By Theorem 7 this happens precisely when the product Π is under-dominant. Otherwise, if it does not converge, it produces upper and lower bounds in (24) that both tend to $\check{\rho}(\mathcal{M})$ as $N \rightarrow \infty$ (Proposition 7). If \mathcal{M} does not have the second invariant cone, then the algorithm can be applied as well, but in some cases it may not converge to the value of LSR. This happens, for instance, when there is a product $A \in \mathcal{M}^r$, whose leading eigenvalue λ_{\max} is unique and simple, and the corresponding eigenvector v has some zero entries. If $[\lambda_{\max}]^{1/r} > [\rho(\Pi)]^{1/n}$ then, for each point v_i produced by the algorithm we have $A^s v_i \rightarrow t_i v$ as $s \rightarrow \infty$, where the sequence $t_i \geq 0$ diverges.

Whence, some vertices of the polytopes P_k converge to the boundary of \mathbb{R}_+^d as $k \rightarrow \infty$. In this case the algorithm does not terminate within finite time, since new vertices v_i will always appear (closer and closer to the boundary of \mathbb{R}_+^d). Moreover, the ratio t_N in (24) may not converge to 1, and the algorithm becomes useless. We suggest the following modification of the algorithm for this case, which often leads to the precise values of LSR.

Assume the algorithm has not terminated after N steps. Take some small $\delta > 0$ and find all vectors $v \in \mathcal{V}_N$ such that $\frac{v_{\min}}{v_{\max}} < \frac{\delta}{d}$, where v_{\min} and v_{\max} are respectively the smallest and the largest entry of v . To any such a vector v we associate a vector h such that $h^q = -\varepsilon$ if $\frac{v^q}{v_{\max}} < \frac{\delta}{d}$, and $h^q = 1$ otherwise (we

write x^q for the q th entree of the vector x). The parameter $\varepsilon > 0$ is chosen to be small and the same for all h . The finite set of all vectors h will be denoted as \mathcal{H} . We also denote by e the vector of ones.

For every $j = 1, \dots, m$ and for every $\bar{h} \in \mathcal{H}$ we solve the following LP problem:

$$\begin{cases} \max & t_e \\ \text{subject to} & \tilde{A}_j \bar{h} \geq t_e e + \sum_{h \in \mathcal{H}} t_h h \\ & \text{and } t_h \geq 0, \quad \forall h \in \mathcal{H}. \end{cases} \quad (25)$$

If for some j and \bar{h} we have $t_e \leq 0$, then for the chosen values of δ and ε the modification is impossible. We can try smaller values. If $t_e > 0$ for all j and $\bar{h} \in \mathcal{H}$, then we restart our algorithm with the same product Π and with the only modification: LP problem (22) is replaced by the following LP problem, where $z = \tilde{A}v$:

$$\begin{cases} \min & t_0 \\ \text{subject to} & t_0 z \geq \sum_{x \in \mathcal{V}_k} t_x x + \sum_{h \in \mathcal{H}} t_h h \\ & \text{and } \sum_{x \in \mathcal{V}_k} t_x \geq 1, \quad t_x \geq 0 \quad \forall x \in \mathcal{V}_k, \quad t_h \geq 0 \quad \forall h \in \mathcal{H}. \end{cases} \quad (26)$$

Explanation. If $t_e > 0$ in LP problem (25) for every $j = 1, \dots, m$ and for every $\bar{h} \in \mathcal{H}$, then the cone $K_{\mathcal{H}} = \{x + \sum_{h \in \mathcal{H}} t_h h \mid x \in \mathbb{R}_+^d, t_h \geq 0, h \in \mathcal{H}\}$ is invariant for the family \mathcal{M} . If the algorithm terminates after the k th step, then the set $P_{k-1} = \text{co}(\mathcal{V}_N) + K_{\mathcal{H}}$ is an extremal infinite polytope for the family \mathcal{M} , i.e., $\tilde{A}P_{k-1} \subset P_{k-1}$ for all $\tilde{A} \in \tilde{\mathcal{M}}$. Therefore, it defines an extremal antinorm in the cone $K_{\mathcal{H}}$, and hence $\check{\rho}(\mathcal{M}) = 1$. Thus, we replace the invariant cone \mathbb{R}_+^d by a wider invariant cone $K_{\mathcal{H}}$, which covers those vertices $v \in \mathcal{V}_N$ that come too close to the boundary of \mathbb{R}_+^d . In many practical cases this trick makes the algorithm converge within finitely many steps. We use it in the proof of Theorem 8 in §8.1.

We consider in the sequel several numerical examples of implementation of our algorithms, and start with the simplest case of nonnegative 2×2 -matrices. In Example 1 Algorithm **(P)** finds the JSR of two matrices, in Example 2 Algorithm **(L)** finds the LSR of another pair of matrices. The aim of those examples is to show how the algorithms work. Then in Section 8 we apply our algorithms to matrices of bigger dimensions (up to $d = 50$) arising in various problems of combinatorics and number theory. Finding the exact values of JSR and LSR we prove, in particular, several previously stated conjectures in combinatorics and number theory, and disprove one conjecture on Pascal's rhombus. Further, in Section 9, we show the statistics how our algorithms work for randomly generated matrices of various dimensions. For all randomly generated families the algorithms found the exact values of JSR and of LSR (the latter is in the case of nonnegative families).

7 Illustrative examples

We consider here some simple examples showing the flow of the algorithms we have presented both for the joint spectral radius and for the lower spectral radius of a nonnegative set of matrices.

Example 1 .Computation of the joint spectral radius. Consider a family $\mathcal{M} = \{A_1, A_2\}$:

$$A_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad A_2 = b \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix},$$

with $b = 9/10$. Looking over all matrix products up to some length, we make a guess that $\Pi = A_1 A_2$ is a spectrum maximizing product. We have $[\rho(\Pi)]^{1/2} = \sqrt{b} \frac{1+\sqrt{5}}{2}$. Applying then Algorithm **(P)** to the family $\tilde{\mathcal{M}} = [\rho(\Pi)]^{-1/2} \mathcal{M}$ we obtain at step zero a unique leading eigenvector v_1 of the product $\tilde{\Pi}_1 = \tilde{\Pi}$, and $v_2 = \tilde{A}_2 v_1$ – the leading eigenvector of the cyclic permutation $\tilde{\Pi}_2$: $v_1 = (1, \frac{\sqrt{5}-1}{2})$ and

$$v_2 = \tilde{A}_2 v_1 = \sqrt{b} \left(\frac{\sqrt{5}-1}{2}, 1 \right) = (0.586318522, 0.948683298)$$

(all the values are rounded to the ninth decimal). At the first step we get one new point

$$v_3 = \tilde{A}_1 v_1 = \frac{1}{\sqrt{b}} \left(1, \frac{\sqrt{5}-1}{\sqrt{5}+1} \right) = (1.054092553, 0.402627528),$$

and the other point $v_4 = \tilde{A}_2 v_2$ is “dead”, because it belongs to the interior of $P_1 = \text{co-}\{v_1, v_2, v_3\}$, i.e., solving LP problem (18) for the point $v = v_2$ and for $\tilde{A} = \tilde{A}_2$ we get $t_{\{v_2, \tilde{A}_2\}} > 1$. Thus, after the first step $\mathcal{V}_1 = \{v_1, v_2, v_3\}$ and $\mathcal{U}_1 = \{v_3\}$.

At the second step we solve LP problem (18) for the pairs (v_3, \tilde{A}_1) and (v_3, \tilde{A}_2) and find that the values $t_{\{v_3, \tilde{A}_1\}}$ and $t_{\{v_3, \tilde{A}_2\}}$ are both bigger than 1. This means that the points $\tilde{A}_1 v_3$ and $\tilde{A}_2 v_3$ are both internal to P_1 . Therefore, $\tilde{A}_j P_1 \subset P_1$, $j = 1, 2$, and so P_1 is an extremal polytope (see Figure 2).

Thus, the algorithm terminates after the second step, and $\hat{\rho}(\mathcal{M}) = [\rho(\Pi)]^{1/2} = \sqrt{b} \frac{1+\sqrt{5}}{2}$.

The cyclic tree of this algorithm is plotted in Figure 3. In Figure 4 we plot the points $\{\tilde{A}_1 v_i\}_{i=1}^3$ (in red) and $\{\tilde{A}_2 v_i\}_{i=1}^3$ (in blue).

Example 2 .Computation of the lower spectral radius. Let $\mathcal{M} = \{A_1, A_2\}$ with

$$A_1 = \begin{pmatrix} 7 & 0 \\ 2 & 3 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 2 & 4 \\ 0 & 8 \end{pmatrix}.$$

We prove that the product $\Pi = A_1 A_2 (A_1^2 A_2)^2$ is spectrum minimizing, and hence the LSR $\check{\rho}(\mathcal{M})$ equals to $\rho(\Pi)^{1/8} = (4(213803 + \sqrt{44666192953}))^{1/8} = 6.009313489\dots$

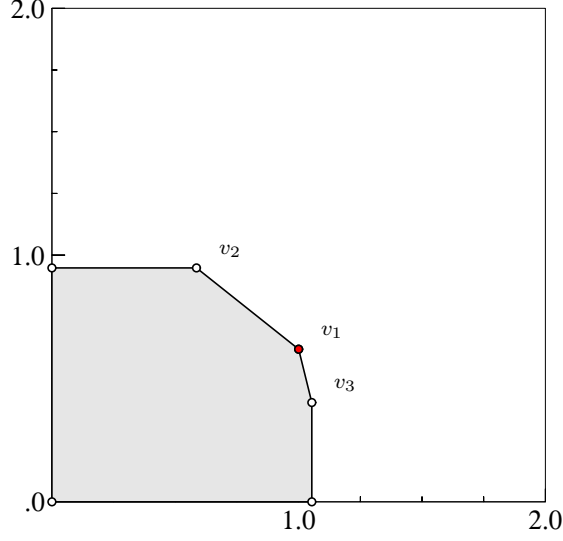


Fig. 2 The extremal polytope P_1 of Example 1. The starting leading eigenvector v_1 of Π is indicated in red.

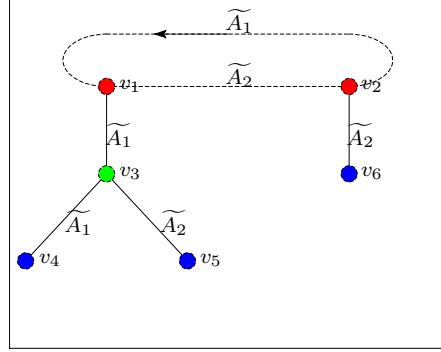


Fig. 3 The cyclic tree of Example 1. The root $\{v_1, v_2\}$ is in red, the alive leaves are in green, the dead leaves are blue.

Starting Algorithm **(L)** we define at zero step $\tilde{M} = [\rho(\Pi)]^{-1/8} \mathcal{M}$, $\tilde{\Pi} = [\rho(\Pi)]^{-1} \Pi$ and get eight points v_1, \dots, v_8 starting from the leading eigenvector

$$v_1 = \left(1, \frac{97444}{\sqrt{44666192953} - 82749} \right) = (1, 0.757760157)$$

of $\tilde{\Pi}$. Thus, $v_2 = \tilde{A}_2 v_1$, $v_3 = \tilde{A}_1 v_2$, $v_4 = \tilde{A}_1 v_3$, $v_5 = \tilde{A}_2 v_4$, $v_6 = \tilde{A}_1 v_5$, $v_7 = \tilde{A}_2 v_6$, $v_8 = \tilde{A}_1 v_7$. At the first step we solve eight LP problems (22) and get the only new alive vertex $v_9 = \tilde{A}_2 v_6$.

The other seven new vertices are “dead leaves”: they belong to the interior of the infinite polytope $P_1 = \text{co}_+ \{v_i\}_{i=1}^9$, for each of them the value $t_{\{v, \tilde{A}\}}$ of problem (22) is smaller than 1. Thus, after the first step we have $\mathcal{V}_1 = \{v_i\}_{i=1}^9$ and $\mathcal{U}_1 = \{v_9\}$. The next step produces two new vertices $\tilde{A}_1 v_9$, $\tilde{A}_2 v_9$ and they are

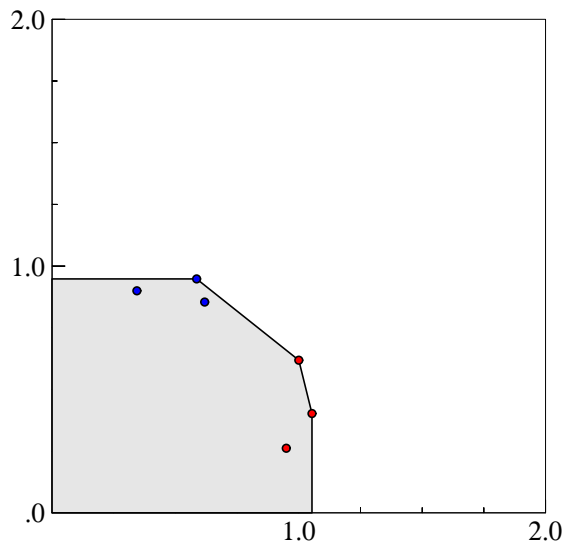


Fig. 4 The extremal polytope of Example 1 and the transformed vectors $\{\tilde{A}_1 v_i\}_{i=1}^3$ (in blue) and $\{\tilde{A}_2 v_i\}_{i=1}^3$ (in red).

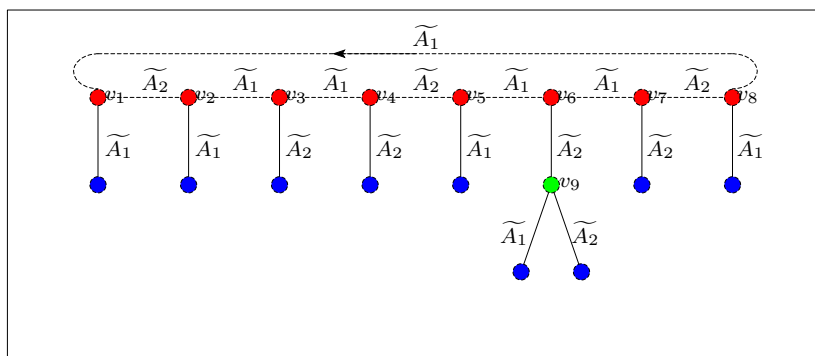


Fig. 5 The cyclic tree of Example 2 (the red root, green alive leaves and blue dead leaves).

both dead. Thus, the algorithm terminates after the second step, and P_1 is the extremal infinite polytope.

8 Applications

We consider four applications of Algorithms **(P)** and **(L)** to various problems of combinatorics, number theory, and theory of formal languages. Each problem is reduced to computing JSR or LSR of some families of nonnegative matrices, which we are able to solve exactly in the sense specified in previous sections.

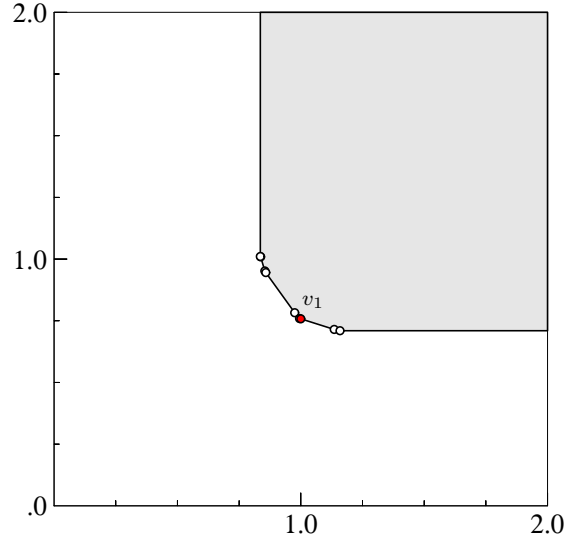


Fig. 6 The extremal infinite polytope in Example 2 (the starting eigenvector v_1 of Π is red).

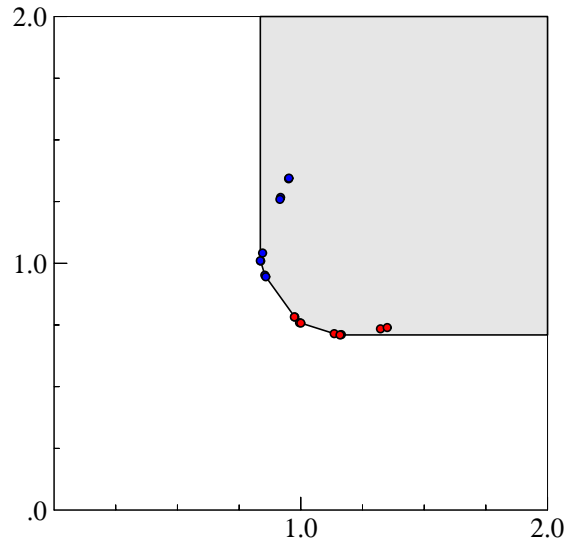


Fig. 7 The extremal infinite polytope P_1 and the transformed vectors $\{\tilde{A}_1 v_i\}_{i=1}^9$ (in red) and $\{\tilde{A}_2 v_i\}_{i=1}^9$ (in blue) of Example 2.

8.1 The asymptotics of the number of overlap-free words

The problem of counting of overlap-free binary words was intensively studied in the literature (see the recent survey [Be]). In [C] and then in [JPB2] this problem was reduced to computing JSR and LSR of two special nonnegative 20×20 -matrices. Those values were computed approximately, and two conjectures were stated about their exact values [JPB2]. Now we prove both those conjectures by applying Algorithms **(P)** and **(L)**.

A binary word, i.e., a finite sequence of zeros and ones, is called *overlap-free* if it does not contain a subword of the form $axaxa$, where $x \in \{0, 1\}$ and a is a word. In 1906 Thue proved that there are infinitely many such words. A natural problem, which was analyzed in many papers, is to estimate the total number u_n of overlap-free words of length n . In 1988 Brlek showed that $u_n \geq 3n - 3$, on the other hand Restivo and Salemi in 1985 proved the polynomial upper bound $u_n \leq C n^r$, where $r = \log(15) \approx 3.906$. This result was sharpened successively by Kfoury (1988), Kobayashi (1988), and Lepistö (1995) to the value $r = 1.37$. On the other hand, Kobayashi (1988) showed that $u_n \geq C n^{1.155}$ (see [Be] for the corresponding references and historical overview). So, the number of overlap-free words grows faster than linearly. A natural question arises, whether $u_n \asymp n^\gamma$ for some $\gamma \in [1.155, 1.37]$. Cassaigne [C] showed that the answer is negative. He introduced the lower and the upper exponents of growth:

$$\begin{aligned}\alpha &= \sup \{r \mid \exists C > 0, u_n \geq C n^r\}, \\ \beta &= \inf \{r \mid \exists C > 0, u_n \leq C n^r\},\end{aligned}\tag{27}$$

and proved that $\alpha < \beta$. Moreover, he established that the numbers u_n can be computed as sums of variables that are obtained by certain linear recurrence relations. This led to the following bounds: $\alpha < 1.276$ and $\beta > 1.332$. The next improvement is due to Jungers, Protasov and Blondel [JPB2], who, showed that $\alpha = \log_2 \check{\rho}(A_1, A_2)$ and $\beta = \log_2 \hat{\rho}(A_1, A_2)$, where A_1 and A_2 are special 20×20 -matrices with nonnegative integer entries, which are reported in the Appendix 10.

In [JPB2] the authors introduced new algorithms for estimating the joint and lower spectral radii based on the convex programming; by means of these algorithms they derived the following bounds:

$$1.2690 < \alpha < 1.2736 \quad \text{and} \quad 1.3322 < \beta < 1.3326.\tag{28}$$

This allowed the authors to make the following conjectures on the precise values:

Conjecture 1 [JPB2]

The s.m.p. for the family \mathcal{M} is $A_1 A_2$, and $\beta = \frac{1}{2} \log_2 \rho(A_1 A_2)$.

Conjecture 2 [JPB2]

The s.l.p. for the family \mathcal{M} is $A_1 A_2^{10}$, and $\alpha = \frac{1}{11} \log_2 \rho(A_1 A_2^{10})$.

Algorithms **(P)** and **(L)** now make it possible to prove both these conjectures.

Theorem 8 For the upper and lower exponents of growth of the function u_n one has:

$$\begin{aligned}\alpha &= \frac{1}{11} \log_2 \rho(A_1 A_2^{10}) = 1.273553265\dots \\ \beta &= \frac{1}{2} \log_2 \rho(A_1 A_2) = 1.332240491\dots\end{aligned}$$

Thus, both the upper and the lower exponents of asymptotic growth of the overlap-free words can be found precisely. To prove Theorem 8 it suffices to present the corresponding extremal polytopes. Since the matrices $A_1 A_2$ are nonnegative, one can apply Algorithm **(P)** for the JSR computation. The candidate for s.m.p. is $\Pi = A_1 A_2$. The algorithm terminates having performed $k = 10$ steps. Thus,

$\widehat{\rho}(A_1, A_2) = \sqrt{\rho(A_1 A_2)} = 2.517934040\dots$ The extremal polytope P_9 has 54 vertices.

To compute the LSR we apply Algorithm **(L)**. The candidate for s.l.p. is $\Pi = A_1 A_2^{10}$. However, performing $k = 10$ steps we see that the algorithm does not converge. There are two sequences of vertices $v_i \in \mathcal{V}_k$ that approach to the boundary of the positive orthant \mathbb{R}_+^d , i.e., those points have very small entries on some positions. Therefore, we apply the modified version of the algorithm (§6.7). Taking $\delta = 1/200$, we see that one sequence have entries of index $q \in I_1 = \{5, 10, 17, 18\}$ smaller than δ , the other sequence have entries of index $q \in I_2 = \{7, 8, 15, 20\}$ smaller than δ . We take $\varepsilon = 1/4$ and $\mathcal{H} = \{h_1, h_2\}$, where $h_j \in \mathbb{R}^{20}$, the q th entry of h_j is $-\varepsilon$ if $q \in I_j$ and $q = 1$ otherwise, $j = 1, 2$. Solving LP problem (25) for all $\bar{h} \in \mathcal{H}$, $A \in \{A_1, A_2\}$ we obtain $t_e > 0$ for each of those four problems, and hence $K_{\mathcal{H}} = \{x + t_1 h_1 + t_2 h_2 \mid x \in \mathbb{R}_+^d, t_1, t_2 \geq 0\}$ is a common invariant cone for A_1, A_2 . Now we apply again Algorithm **(L)** with the cone $K_{\mathcal{H}}$ instead of \mathbb{R}_+^d , i.e., replacing LP problem (22) by (26). The modified algorithm converges: it terminates after $k = 15$ steps. Thus, $\Pi = A_1 A_2^{10}$ is an s.l.p., and $\check{\rho}(A_1, A_2) = [\rho(A_1 A_2^{10})]^{1/11} = 2.417562630\dots$ The extremal infinite polytope P_{14} has 104 vertices.

In order to give the formal proof of the theorem it is sufficient to simply provide the list of vertices of the corresponding extremal polytope P_9 (to compute JSR) and of the extremal infinite polytope P_{14} (to compute LSR). They are given in the Appendix 10.

8.2 The density of ones in the Pascal rhombus

The Pascal rhombus is related to the Pascal triangle, with the only difference that each element equals to the sum of four previous elements rather than two (see [GKMT] for definitions and basic properties). The elements of the Pascal rhombus arise from linear recurrence relations on polynomials. The sequence of polynomials $\{p_n\}$ is defined as $p_0(x) = 1$, $p_1(x) = x^2 + x + 1$ and $p_n(x) = (x^2 + x + 1)p_{n-1}(x) + x^2 p_{n-2}(x)$, $n \geq 2$. This leads to a recurrence relation for the number w_n of odd coefficients of p_n [FSB]. The asymptotic growth of w_n as $n \rightarrow \infty$ is characterized as follows:

$$\limsup_{n \rightarrow \infty} \frac{\log w_n}{\log n} = \log_2 \widehat{\rho}; \quad \liminf_{n \rightarrow \infty} \frac{\log w_n}{\log n} = \log_2 \check{\rho},$$

where $\widehat{\rho}$ and $\check{\rho}$ are respectively the JSR and LSR of matrices A_1, A_2 defined as

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (29)$$

This is shown easily that $\widehat{\rho} = 2$, and the main difficulty is to compute $\check{\rho}$.

Conjecture 3 [F] For the set of matrices (29) we have $\check{\rho} = \frac{\sqrt{5}+1}{2} = 1.61803\dots$

This conjecture is quite natural, because the golden number appears in many problems of combinatorics. An approximate computation of $\check{\rho}(A_1, A_2)$ given in [PJB] provided the following estimate:

$$1.6180 \leq \check{\rho} \leq 1.6376, \quad (30)$$

which rather confirms Conjecture 3. The upper bound in (30) is obtained by the product $A_1^3 A_2^3$, for which $\rho^{1/6}(A_1^3 A_2^3) = 1.6376\dots$

Applying Algorithm (L) we find the precise value of $\check{\rho}$. It follows that Conjecture 3 is not true, and the product $A_1^3 A_2^3$ is actually an s.l.p.

Theorem 9 *For the family $\mathcal{M} = \{A_1, A_2\}$ one has $\check{\rho}(\mathcal{M}) = \rho^{1/6}(A_1^3 A_2^3) = 1.6376\dots$*

This solves the problem of asymptotics of the sequence w_n and disproves the golden number conjecture.

Proof of Theorem 9. To prove the theorem it is convenient to apply Algorithm (L) to the transposed family $\mathcal{M}^* = \{A_1^*, A_2^*\}$ rather than to original family (29). Of course, $\check{\rho}(\mathcal{M}^*) = \check{\rho}(\mathcal{M})$. We take $\Pi = (A_1^*)^3 (A_2^*)^3$ as a candidate for s.l.p. The leading eigenvector of $\tilde{\Pi}_1 = \tilde{\Pi}$ is

$$v_1 = (0.39925900, 0.95496725, 0.79851800, 0.90909090, 1).$$

At step zero we get six vertices: $v_2 = \tilde{A}_2^* v_1$, $v_3 = \tilde{A}_2^* v_2$, $v_4 = \tilde{A}_2^* v_3$, $v_5 = \tilde{A}_1^* v_4$, $v_6 = \tilde{A}_1^* v_5$. The first step gives two new alive vertices: $v_7 = \tilde{A}_1^* v_3$ and $v_8 = \tilde{A}_2^* v_6$. The algorithm terminates after the second step, since we compute that the images of the new vertices v_7, v_8 lie inside the infinite polytope $P_1 = \text{co}_+\{v_1, \dots, v_8\}$. Thus, we obtain the leading eigenvectors of all cyclic permutations of $\tilde{\Pi}$ plus two additional points (v_7 and v_8). One can check directly that the infinite polytope $P_1 = \text{co}_+\{v_1, \dots, v_8\}$, where the vertices $\{v_i\}_{i=1}^8$ are described above, is extremal for $\tilde{\mathcal{M}}^*$, i.e., $\tilde{A}_j^* P_1 \subset P_1$, $j = 1, 2$.

Thus, the polytope P_1 is extremal, and the product $(A_1^*)^3 (A_2^*)^3$ is an s.l.p. In Figure 8 we see the cyclic tree of P_1 . Moreover, by Theorem 7 the product $(A_1^*)^3 (A_2^*)^3$ is under-dominant. Hence, the transpose product $A_2^3 A_1^3$ is under-dominant for \mathcal{M} , and therefore so is the product $A_1^3 A_2^3$ being its cyclic permutation.

This concludes the proof. □

8.3 The Euler binary partition function

For an arbitrary integer $r \geq 2$ the Euler binary partition function $b(k) = b_2(r, k)$ is defined on the set of nonnegative integers k as the total number of different binary expansions $k = \sum_{j=0}^{\infty} d_j 2^j$, where the "digits" d_j take values from the set $\{0, \dots, r-1\}$. The asymptotic behavior of $b(k)$ as $k \rightarrow \infty$ was studied in various interpretations by L. Euler, K. Mahler, N.G. de Bruijn, D.E. Knuth, B. Reznick and others (see [P3] for the corresponding references). For even $r = 2n$, as it was

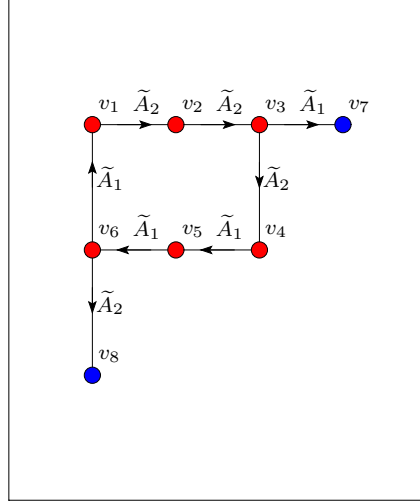


Fig. 8 The vertices $\{v_i\}_{i=1}^8$ for LSR computation of the Pascal rhombus.

r	JSR				LSR			
	# its	# vertices	$\hat{\rho}$	s.m.p.	# its	# vertices	$\hat{\rho}$	s.l.p.
7	5	8	3.511547	A_1	6	14	3.491891	$A_1 A_2$
9	6	18	4.503099	$A_1 A_2$	5	17	4.494492	A_1
11	5	14	5.505892	A_1	7	24	5.497042	$A_1 A_2$
13	5	16	6.502167	A_1	7	28	6.498946	$A_1 A_2$
15	7	40	7.500106	$A_1 A_2$	6	23	7.499841	A_1
17	7	40	8.500057	$A_1 A_2$	6	30	8.499904	A_1
19	7	24	9.500423	A_1	8	46	9.499789	$A_1 A_2$
21	6	28	10.500373	A_1	8	50	10.499813	$A_1 A_2$
23	8	52	11.500053	$A_1 A_2$	6	31	11.499894	A_1
25	9	34	12.500059	A_1	8	58	12.499971	$A_1 A_2$
27	8	60	13.500030	$A_1 A_2$	7	37	13.499938	A_1
29	9	66	14.500009	$A_1 A_2$	8	43	14.499982	A_1
31	9	30	15.500001	A_1	10	34	15.499999	$A_1 A_2$
33	11	36	16.500001	$A_1 A_2$	10	55	16.499999	A_1
35	8	52	17.500007	A_1	18	102	17.499997	$A_1 A_2$
37	8	54	18.500012	A_1	10	113	18.499994	$A_1 A_2$
39	10	112	19.500003	$A_1 A_2$	8	59	19.499994	A_1
41	9	78	20.500005	A_1	11	120	20.499997	$A_1 A_2$

Table 1 Computation of the JSR and of the LSR for the Euler partition function matrices.

shown in [R], one has $b(k) \asymp k^{\log_2 n}$. For odd values of r the asymptotic behavior of $b(k)$ is more complicated and has been studied in [R] and [P3]. Denote

$$p_1 = \liminf_{k \rightarrow \infty} \log b(k) / \log k; \quad p_2 = \limsup_{k \rightarrow \infty} \log b(k) / \log k. \quad (31)$$

In [P3] it was proved that $p_1 = \log_2 \tilde{\rho}(A_1, A_2)$ and $p_2 = \log_2 \hat{\rho}(A_1 A_2)$, where A_1, A_2 are $(r-1) \times (r-1)$ -matrices defined as follows: $(A_s)_{ij} = 1$ if $2-s \leq 2j-i \leq r-s+1$, and $(A_s)_{ij} = 0$ otherwise (for $s = 1, 2$). For example,

for $r = 7$ we have the following 6×6 -matrices:

$$A_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}; \quad A_2 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

In [P3] the following conjecture was made:

Conjecture 4 *For every odd r one of the two products A_1 and $A_1 A_2$ is an s.m.p. and the other is an s.l.p.*

The case $r = 3$ was carried out earlier in the work [R], for $r = 5, 7, 9, 11, 13$ Conjecture 4 was proved to hold true in [P3]. Algorithms **(P)** and **(L)** make it possible to prove this conjecture for many more odd values (in particular we did the computation for $r \leq 41$).

The results are listed in Table 1. The first column is r (where we recall that the dimension of matrices A_1, A_2 is $r - 1$), the second column is the number k of iterations necessary to Algorithm **(P)** for terminating, the third column is the number of vertices of the extremal polytope P_{k-1} , the fourth one is the value of JSR rounded to the sixth decimal, and the fifth one is the s.m.p. The right hand side of the table presents analogous informations for the LSR computation by Algorithm **(L)**.

We see that Algorithms **(P)** and **(L)** demonstrate a good efficiency. Even for large dimensions of the matrices A_1, A_2 the total number of iterations k never exceeds 18 and the number of vertices of the extremal polytope P_{k-1} is at most 120. Let us remark that the binary matrices A_1, A_2 of the partition function are rather inconvenient for our algorithms, because of a very small gap between JSR and LSR. For instance, for $r = 33$ the distinction between the JSR and LSR is less than 0.00002%. Therefore, all products of A_1 and A_2 of some length k have almost the same spectral radii. This is why we would expect Algorithms **(P)** and **(L)** to need a large number of iterations. On the contrary, they just need 11 and 10 iterations respectively.

Actually Algorithms **(P)** and **(L)** work also for higher dimensions and Conjecture 4 can be proved for larger r . For instance, if $r = 51$, then Algorithm **(L)** needs $k = 15$ iterations and produces an extremal infinite polytope P_{14} with 135 vertices.

In the next section, as further example, we consider ternary expansions and show that also in this case we can compute the significant measures.

8.4 The Euler ternary partition function

The LSR and JSR appear in the problem of asymptotics of the Euler partition function on the arbitrary base, not only for binary expansions. For instance, the ternary partition function $b(k) = b_3(r, k)$ is the total number of different ternary expansions $k = \sum_{j=0}^{\infty} d_j 3^j$, where the "digits" d_j take values from the set $\{0, \dots, r-1\}$. The largest and the smallest exponents of growth of $b(k)$ as $k \rightarrow \infty$ are defined by

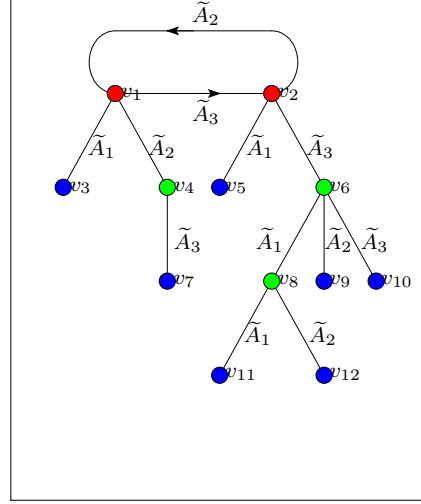


Fig. 9 The vertices $\{v_i\}_{i=1}^{12}$ for JSR computation of the ternary Euler partition function.

formulas similar to (31) through the LSR and the JSR of three special binary matrices A_1, A_2, A_3 (see [P3] for details). In [PJB] the authors analyze the example with $r = 14$, where the matrices of $\mathcal{M} = \{A_1, A_2, A_3\}$ are

$$A_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

In [PJB] the values $\check{\rho}(\mathcal{M})$ and $\hat{\rho}(\mathcal{M})$ were computed approximately to the following accuracy:

$$4.525 \leq \check{\rho}(\mathcal{M}) \leq 4.6105; \quad 4.72 \leq \hat{\rho}(\mathcal{M}) \leq 4.8.$$

Algorithms **(P)** and **(L)** determine their precise values:

$$\check{\rho}(\mathcal{M}) = [\rho(A_1 A_2)]^{1/2} = 4.61047781 \dots$$

$$\hat{\rho}(\mathcal{M}) = [\rho(A_2 A_3)]^{1/2} = 4.72204513 \dots$$

The JSR computation. Algorithm **(L)** starting with the product $\Pi = A_2 A_3$ terminates after 4 steps producing the extremal infinite polytope $P_3 = \text{co-}\{v_i\}_{i=1}^{12}$,

where v_1 is the leading eigenvector of \tilde{H} , $v_2 = \tilde{A}_3 v_1$ (step zero); $v_3 = \tilde{A}_1 v_1$, $v_4 = \tilde{A}_2 v_1$, $v_5 = \tilde{A}_1 v_2$, and $v_6 = \tilde{A}_3 v_2$ (first step); $v_7 = \tilde{A}_3 v_4$, $v_8 = \tilde{A}_1 v_6$, $v_9 = \tilde{A}_2 v_6$, and $v_{10} = \tilde{A}_3 v_6$ (second step); $v_{11} = \tilde{A}_1 v_8$ and $v_{12} = \tilde{A}_2 v_8$ (third step). See the corresponding cyclic tree in Figure 9. Thus $\Pi = A_2 A_3$ is an s.m.p.

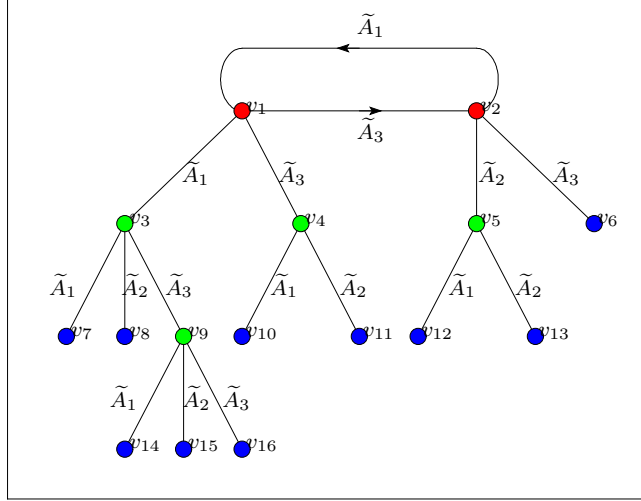


Fig. 10 The vertices $\{v_i\}_{i=1}^{16}$ for LSR computation of the ternary Euler partition function

The LSR computation. Algorithm (P) starting with the product $\Pi = A_1 A_2$ terminates after 4 steps, the extremal polytope $P_3 = \text{co}_+\{v_i\}_{i=1}^{16}$. The vertex v_1 is the leading eigenvector of \tilde{H} , $v_2 = \tilde{A}_2 v_1$ (step zero); $v_3 = \tilde{A}_1 v_1$, $v_4 = \tilde{A}_3 v_1$, $v_5 = \tilde{A}_2 v_2$, and $v_6 = \tilde{A}_3 v_2$ (first step); $v_7 = \tilde{A}_1 v_3$, $v_8 = \tilde{A}_2 v_3$, $v_9 = \tilde{A}_3 v_3$, $v_{10} = \tilde{A}_1 v_4$, $v_{11} = \tilde{A}_2 v_4$, $v_{12} = \tilde{A}_1 v_5$, and $v_{13} = \tilde{A}_2 v_5$ (second step); $v_{14} = \tilde{A}_1 v_9$, $v_{15} = \tilde{A}_2 v_9$, and $v_{16} = \tilde{A}_3 v_9$ (third step). See the corresponding cyclic tree in Figure 10. Therefore $\Pi = A_1 A_2$ is an s.l.p.

9 Numerical results for randomly generated matrices

In this section we report some results obtained for families consisting of a pair of random matrices of variable dimensions d .

The results show that the computation complexity increases significantly as the dimension increases but also confirm the effectiveness of the method for computing the joint and the lower spectral radius of nonnegative matrices. We expect in general that the reachable dimension for a computation in a reasonable time might be quite high for a set of operators sharing an invariant cone.

First we consider the general case of two random matrices with normally distributed entries. The generated random matrices are scaled to have equal spectral norm. This aims to reduce the number of cases where the s.m.p is the matrix with larger spectral radius. The first column of Table 2 gives the dimension, the second

JSR				JSR			
d	# its	# vertices	s.m.p.	d	# its	# vertices	s.m.p.
5	3	14	$A_1 A_2$	6	4	26	A_1
5	7	23	$A_1 A_2^2$	6	9	51	$A_1 A_2$
5	12	37	A_1	6	5	38	$A_1^2 A_2$
7	17	100	A_1	8	19	117	$A_1^3 A_2 A_1^4 A_2$
7	12	140	$A_1^3 A_2 A_1 A_2$	8	8	49	A_1
7	24	223	$A_1^3 A_2^2$	8	12	75	$A_1 A_2^3$
9	18	177	$A_1^8 A_2$	10	16	239	$A_1 A_2^4$
9	13	172	$A_1^3 A_2 A_1 A_2$	10	9	109	A_1
9	10	129	A_2	10	24	408	$(A_1^3 A_2)^2 A_2$
11	20	707	$A_1^3 A_2^2$	12	31	1539	$A_1 A_2 A_1^2 A_2^2$
11	14	340	$A_1^2 A_2 A_1 A_2$	12	9	211	$A_1 A_2$
11	12	183	$A_1^3 A_2$	12	13	215	$A_1 A_2^3$
15	18	715	$A_1^2 A_2 A_1 A_2^4$	20	21	1539	$A_1 A_2$
15	14	570	$A_1^4 A_2$	20	16	1219	$A_1 A_2^2$
15	14	390	A_2	20	16	1247	$A_1^2 A_2^2$

Table 2 Computation of the JSR for random pairs of matrices with equal norm.

column the number of iterations for Algorithm **(R)** to converge, the third column provides the number of vertices of the extremal polytope and the last column gives the correspondent s.m.p.; we immediately observe that the complexity (in terms of iterations and number of vertices) rapidly increases with the dimension. Dealing with two 20×20 matrices can be considered a challenging computational problem.

Then we consider in Table 3 randomly generated nonnegative matrices still scaled to have the same norm.

JSR				LSR		
d	# its	# vertices	s.m.p.	# its	# vertices	s.l.p.
10	3	6	$A_1 A_2$	4	6	$A_1 A_2^2$
10	3	4	A_1	4	5	A_2
10	4	6	$A_1 A_2$	7	15	$A_1^2 A_2^2$
10	6	11	$A_1^2 A_2^2$	3	6	$A_1 A_2$
10	4	8	$A_1 A_2^2$	5	9	$A_1^2 A_2$
20	4	7	A_2	4	6	A_1
20	4	6	$A_1 A_2$	5	9	$A_1 A_2^2$
20	6	14	$A_1^2 A_2$	3	4	$A_1 A_2$
20	5	11	$A_1 A_2^2$	6	14	$A_1^2 A_2$
20	5	9	$A_1 A_2$	3	4	A_1

Table 3 Computation of the JSR and of the LSR for random nonnegative pairs of matrices.

Finally we consider binary matrices and vary the density of the number of zero entries. We scale the pairs of matrices to have the same spectral radius; note that

in some cases either the s.m.p. or the s.l.p. are the starting matrices so that there is no guarantee of the convergence of the algorithm we propose. Nevertheless, when we report the number of iterations and of vertices we imply that the algorithm has converged in a finite number of steps.

Tables 4 and 5 report the results obtained for pairs of matrices respectively of dimension $d = 50$ and $d = 100$. Whenever both A_1 and A_2 are either s.m.p.'s or s.l.p.'s we indicate both the numbers of iterations/vertices taking either A_1 or A_2 as optimal product.

density	JSR			LSR		
	# its	# vertices	s.m.p.	# its	# vertices	s.l.p.
0.2	9	55	$A_1 A_2^2$	4	8	A_1 and A_2
0.2	5	17	$A_1 A_2$	5	10	$A_1^2 A_2$
0.2	8	24	$A_1^2 A_2^2$	4 (4)	6 (6)	A_1 and A_2
0.2	5	16	$A_1^2 A_2$	4 (5)	6 (8)	A_1 and A_2
0.2	14	59	$A_1 A_2^3$	5	10	$A_1 A_2$
0.5	4	8	$A_1 A_2$	4	10	$A_1 A_2$
0.5	5	14	$A_1^2 A_2$	4 (3)	5 (4)	A_1 and A_2
0.5	6	15	$A_1 A_2^2$	6	17	$A_1^2 A_2^2$
0.5	5	16	$A_1 A_2$	4 (4)	6 (5)	A_1 and A_2
0.5	6	20	$A_1^3 A_2$	5	9	$A_1 A_2$
0.75	5	16	$A_1 A_2^2$	5 (7)	12 (14)	A_1 and A_2
0.75	4	8	$A_1 A_2$	6	16	$A_1 A_2$
0.75	5	11	$A_1 A_2^3$	6	19	$A_1 A_2^2$
0.75	5	16	$A_1^2 A_2^2$	11	170	$A_1 A_2^5$
0.75	5	12	$A_1^2 A_2$	5 (6)	13 (12)	A_1 and A_2
0.9	4 (5)	8 (9)	A_1 and A_2	4	8	$A_1 A_2$
0.9	5	9	$A_1^2 A_2$	6	4	$A_1 A_2$
0.9	3	4	$A_1 A_2$	7 (8)	11 (12)	A_1 and A_2
0.9	4	11	$A_1 A_2^2$	4	7	$A_1 A_2$
0.9	7	14	$A_1^3 A_2^2$	8 (8)	13 (11)	A_1 and A_2

Table 4 Computation of the JSR and of the LSR for random pairs of binary matrices of dimension $d = 50$.

Some comments are necessary. The computations for the general case with $d = 10$ need usually a few minutes. The computations for the general case with $d = 20$ need usually between half an hour and one hour of computation but for some examples till 8 hours (in a standard laptop with i5 processor). We hypothesize that the overall computation depend on several factors, not only the length of the spectrum maximizing product but also the ratio between the leading eigenvalues of the family and the closer ones that is eigenvalues of products which have modulus close to 1 and on the distribution of the vertices of the extremal polytope in all the orthants.

In the nonnegative case all the vertices lie in the nonnegative orthant and this determine a much lower complexity. The presence of quasi-optimal products that is products with eigenvalues close to 1 is a factor of slowdown also in this case.

density	JSR			LSR		
	# its	# vertices	s.m.p.	# its	# vertices	s.l.p.
0.2	6	24	$A_1^2 A_2$	6	31	$A_1 A_2^2$
0.2	6	23	$A_1 A_2$	6	28	$A_1^2 A_2^2$
0.2	7	27	$A_1 A_2^3$	6	20	$A_1 A_2$
0.2	5	21	$A_1 A_2^2$	7	24	$A_1^2 A_2$
0.5	5	10	$A_1 A_2$	5	15	$A_1 A_2^2$
0.5	6	17	$A_1^2 A_2$	4	8	$A_1 A_2$
0.5	6	18	$A_1^2 A_2^2$	5	16	$A_1^2 A_2$
0.5	6	22	$A_1 A_2^3$	4 (6)	9 (14)	A_1 and A_2
0.8	4	7	$A_1 A_2$	4	7	$A_1 A_2$
0.8	7	18	$A_1^2 A_2$	6	14	$A_1^2 A_2^2$
0.8	5	14	$A_1 A_2^2$	9 (7)	14 (16)	A_1 and A_2
0.8	5	12	$A_1^3 A_2$	5	12	$A_1 A_2^2$

Table 5 Computation of the JSR and of the LSR for random pairs of binary matrices of dimension $d = 100$.

For binary matrices we observe from the experiments that the behavior of the algorithm slightly depends on the density of the zero entries and also on the dimension. This implies that we are able to compute the joint spectral characteristics of possibly large binary matrices with any density.

In some applications the families of the matrices have the same spectral radii and it may happen that the optimal products are exactly the matrices themselves. In our experience there are cases where we have been able to compute an extremal polytope invariant set by starting from any optimal matrix of the family (see also the tables of results) but we have also encountered cases where the algorithm has not terminated finitely. We think that an interesting open problem is that of balancing leading eigenvectors associated to different products (which are not powers or cyclic permutation one of the other).

Although it is true that this situation is not generic there are some applications where it naturally occurs. We leave this topic to a future investigation.

10 Appendix

We give here the proofs of two main results of this paper (Theorems 4 and 7) and details about the proof of Theorem 8.

10.1 Proof of Theorem 4.

We give the proof for Algorithm **(R)**, the proofs for Algorithms **(C)** and **(P)** are analogous. We use two auxiliary results.

Lemma 12 *Let us have a cyclic tree \mathcal{T} with a root \mathbf{B} generated by an irreducible word $\mathbf{b} = d_n \dots d_1$; then for any word \mathbf{a} , which is not a power of \mathbf{b} , we have $\mathbf{a}^n \notin \mathbf{B}$.*

Proof. Let l be the length of \mathbf{a} and p be and the greatest common divisor of l and n . If $l = kn$ for some integer k , then the words \mathbf{a} and \mathbf{b}^k must have different letters at some position, otherwise $\mathbf{a} = \mathbf{b}^k$. Therefore $\mathbf{a} \notin \mathbf{B}$, and so $\mathbf{a}^n \notin \mathbf{B}$.

If l is not divisible by n , then $p < n$, and there exists an index j such that $d_{j+p} \neq d_j$, otherwise \mathbf{b} is a power of the word $d_p \dots d_1$, which contradicts the irreducibility. The Diophantine equation $lx - ny = p$ has a solution $(x, y) \in \mathbb{Z}^2$ such that $0 \leq x \leq n - 1$. Since the words \mathbf{a}^{x+1} and \mathbf{b}^{y+2} have different letters at the position $lx + j$, we have $\mathbf{a}^{x+1} \notin \mathbf{B}$, and hence $\mathbf{a}^n \notin \mathbf{B}$, because $n \geq x + 1$. \square

Lemma 13 *Let \mathcal{T} be the cyclic tree generated by the product $\tilde{\Pi}$. If Algorithm (R) terminates within k steps, then there is $\varepsilon > 0$ such that all nodes of \mathcal{T} of level $\geq k$ are in the polytope $(1 - \varepsilon) P_{k-1}$.*

Proof. If the algorithm terminates after k steps, then $\tilde{A}_j P_{k-1} \subset P_{k-1}$ for all $j = 1, \dots, m$. Moreover we have $\mathcal{U}_k = \emptyset$, which means that every node $v \in \mathcal{T}$ of the level k belongs to a dead branch, and therefore $v \in \text{int } P_{k-1}$. The total number of nodes of level k is finite, hence all of them are in $(1 - \varepsilon) P_{k-1}$ for some $\varepsilon > 0$. If v is a node of a bigger level $r > k$, then $v = R v_0$, where $R \in \tilde{\mathcal{M}}^{r-k}$ and v_0 is some node of the k th level. Since $v_0 \in (1 - \varepsilon) P_{k-1}$, we have $v \in (1 - \varepsilon) R P_{k-1} \subset (1 - \varepsilon) P_{k-1}$, because $R P_{k-1} \subset P_{k-1}$. \square

Proof of Theorem 4.

Necessity. Consider the cyclic tree \mathcal{T} generated by the product $\tilde{\Pi}$. Assume the algorithm terminates after k steps. By Lemma 13 all nodes of levels at least k belong to $(1 - \varepsilon) P_{k-1}$, where $\varepsilon > 0$ is fixed. For every product C , which is not a power of $\tilde{\Pi}_i$, the node $C^n v_i$ does not belong to the root (Lemma 12). Hence for each v_i from the root and for every product $C \in \tilde{\mathcal{M}}^l$ that is not a power of a cyclic permutation of $\tilde{\Pi}$, the level of the node $C^{n+k} v_i$ is bigger than k . If v is not in the root, then this level is bigger than $ln + k > k$. Thus, $C^{n+k} v \in (1 - \varepsilon) P_{k-1}$ for each node $v \in \mathcal{T}$, and hence, for each vertex v of the polytope P_{k-1} . This yields that $C^{n+k} P_{k-1} \subset (1 - \varepsilon) P_{k-1}$. Therefore, $\rho(C^{n+k}) < 1 - \varepsilon$, and so $\rho(C) < (1 - \varepsilon)^{1/(n+k)}$. Consequently $\tilde{\Pi}$ is dominant.

Let us now show that 1 is its unique and simple leading eigenvalue. Since for $i \neq 1$ the product $\tilde{\Pi}_1$ is not a power of $\tilde{\Pi}_i$, it follows that the node $\tilde{\Pi}_1^n v_i$ does not belong to the root (Lemma 13). Hence, the level of the node $\tilde{\Pi}_1^{n+k} v_i$ is bigger than k . If v is not in the root, then the level of $\tilde{\Pi}_1^{n+k} v$ is bigger than k as well. Thus, $\tilde{\Pi}_1^{n+k} v \in (1 - \varepsilon) P_{k-1}$ for all vertices v of P_{k-1} , except for $v = \pm v_1$. For any eigenvector $u \neq v_1$ of the operator $\tilde{\Pi}_1$ take the one-dimensional subspace $U \subset \mathbb{R}^d$ spanned by u (if u is complex, then U is the two-dimensional subspace spanned by u and by its conjugate). Since $v_1 \notin U$, it follows that $\tilde{\Pi}_1(P_{k-1} \cap U) \subset \text{int}(P_{k-1} \cap U)$, where the interior is taken in U . This implies that the spectral radius of $\tilde{\Pi}_1$ on the subspace U is smaller than 1. Thus, all eigenvalues of $\tilde{\Pi}_1$ different from 1 are smaller than 1 by modulo, and the eigenvalue 1 has a unique eigenvector. Hence, the leading eigenvalue 1 is unique and has only one Jordan block. The dimension of this block cannot exceed one, otherwise $\|\tilde{\Pi}_1^k\| \rightarrow \infty$ as $k \rightarrow \infty$, which contradicts the nondefectivity of the family $\tilde{\mathcal{M}}$. Therefore, the eigenvalue 1 is simple.

Sufficiency. The proof uses similar arguments as this given for the *Small CPE Theorem* in [GWZ], to which we refer the reader.

Assume $\tilde{\Pi}$ is dominant and its leading eigenvalue is unique and simple. If the algorithm does not terminate, then the tree \mathcal{T} has an infinite path of alive leaves $v^{(0)} \rightarrow v^{(1)} \rightarrow \dots$ (the node $v^{(i)}$ is on the i th level) starting at a node $v^{(0)} = v_p$ from the root. For every r we have $v^{(r)} \notin \text{int } P_{r-1}$. Hence $v^{(r)} \notin \text{int } P_k$ for all $k < r$. Since the family $\tilde{\mathcal{M}}$ is irreducible, it follows that $0 \in \text{int } P_k$ for some k , and hence the polytope P_k defines a norm $\|\cdot\|_k$ in \mathbb{R}^d . For this norm $\|v^{(r)}\| \geq 1$ for all $r > k$. On the other hand, $\tilde{\mathcal{M}}$ is nondefective, hence the sequence $\{v^{(r)}\}$ is bounded. Thus, there is a subsequence $\{v^{(r_i)}\}_{i \in \mathbb{N}}$, $r_1 \geq k$, that converges to some point $v \in \mathbb{R}^d$. Clearly, $\|v\|_k \geq 1$. For every i we have $v^{(r_i)} = R_i v^{(r_1)}$ and $v^{(r_{i+1})} = C_i v^{(r_i)}$, where $R_i \in \tilde{\mathcal{M}}^{r_i - r_1}$, $C_i \in \tilde{\mathcal{M}}^{r_{i+1} - r_i}$. Denote by $\text{cl}[\tilde{\mathcal{M}}]$ the closure of the semigroup of all products of operators from $\tilde{\mathcal{M}}$. Since this semigroup is bounded, after possible passage to a subsequence, it may be assumed that R_i and C_i converge to some $R, C \in \text{cl}[\tilde{\mathcal{M}}]$ respectively as $i \rightarrow \infty$. We have $Cv = v$, hence $\rho(C) \geq 1$, which, by the domination assumption, implies that there is $j \in \{1, \dots, n\}$ such that C belongs to $\text{cl}[\tilde{\Pi}_j]$, which is the closure of the semigroup $\{(\Pi_j)^q\}_{q \in \mathbb{N}}$. Moreover, since the leading eigenvalue of $\tilde{\Pi}$ is unique and simple, we see that $v = \lambda v_j$, where $\lambda \in \mathbb{R}$. We have, $\|v_j\|_k = 1$ and $\|v\|_k \geq 1$, hence $|\lambda| \geq 1$. Thus, $Rv^{(r_1)} = \lambda v_j$. The nodes v_j and $v^{(0)} = v_p$ are both from the root, hence there is a product S such that $Sv_j = v^{(1)}$. Taking into account that $v^{(r_1)} = R_1 v^{(1)}$, we obtain $R_1 S R v^{(1)} = \lambda v^{(1)}$. Hence $\rho(R_1 S R) \geq |\lambda|$, and we conclude that $\lambda = \pm 1$ and that $R_1 S R \in \text{cl}[\tilde{\Pi}_i]$ for some i . This yields $v^{(1)} = \mu v_i$, where v_i is the corresponding vector from the root, $\mu \in \mathbb{R}$. Since $\|v_i\| = 1$ and $\|v^{(1)}\| \geq 1$, we have $|\mu| \geq 1$. The elements v_i and $v^{(0)} = v_p$ are both from the root, hence $\tilde{A}_p \cdots \tilde{A}_i v_i = v_p$, and consequently $\tilde{A}_s \tilde{A}_p \cdots \tilde{A}_i v_i = v^{(1)}$ for some $\tilde{A}_s \in \tilde{\mathcal{M}}$. Note that $d_s \neq d_{p+1}$, because the node $v^{(1)}$ is not in the root. Therefore, the product $Q = \tilde{A}_s \tilde{A}_p \cdots \tilde{A}_i$ does not coincide with Π_i , and its length is at most n . Thus, $Q v_i = \mu v_i$, hence $\rho(Q) = 1$ and $\mu = \pm 1$. Thus, Q has spectral radius 1 and the leading eigenvector v_i , therefore $Q \in \text{cl}[\tilde{\Pi}_i]$. On the other hand the length of Q does not exceed n , hence $Q = \tilde{\Pi}_i$, which is a contradiction. Hence, the algorithm terminates within finitely many steps. \square

10.2 Proof of Theorem 7.

We use several auxiliary results. The proof of the following lemma is similar to the proof of Lemma 13.

Lemma 14 *Let \mathcal{T} be the cyclic tree generated by the product $\tilde{\Pi}$. If the algorithm terminates within k steps, then there is $\varepsilon > 0$ such that all vertices of the tree of level $\geq k$ belong to the infinite polytope $(1 + \varepsilon) P_{k-1}$.*

Lemma 15 [Va] *If an operator B has an invariant cone K , then for every its eigenvector from $\text{int } K$ the corresponding eigenvalue equals to $\rho(B)$.*

Proof of Theorem 7.

Necessity. Consider the cyclic tree \mathcal{T} generated by the product \tilde{H} . If the algorithm terminates after k steps, then by Lemma 14 all vertices of levels at least k belong to $(1 + \varepsilon)P_{k-1}$. For an arbitrary product C , which is not a power of \tilde{H}_i , the node $C^n v_i$ does not belong to the root (Lemma 12). Hence for every v_i from the root the level of the node $C^{n+k} v_i$ is bigger than k . If v is not in the root, then the level of $C^{n+k} v$ is bigger than k as well, consequently $C^{n+k} v \in (1 + \varepsilon)P_{k-1}$ for each node $v \in \mathcal{T}$, and hence for every vertex of P_k . This yields that $C^{n+k} P_{k-1} \subset (1 + \varepsilon)P_{k-1}$, therefore $\rho(C^{n+k}) > 1 + \varepsilon$, and so $\rho(C) > (1 + \varepsilon)^{1/(n+k)}$. This holds for every product C that is not a power of \tilde{H} or of its cyclic permutations, which completes the proof.

Sufficiency. Assume the converse: the product \tilde{H} is under-dominant, but the algorithm does not produce an extremal infinite polytope. This means that the tree \mathcal{T} has an infinite path of alive leaves $v^{(0)} \rightarrow v^{(1)} \rightarrow \dots$ starting at a vertex $v^{(0)} = v_p$ from the root. Since the family \mathcal{M} is eventually positive, it follows from Lemma 9 that there exists an internal invariant cone \tilde{K} , which, moreover, contains all leading eigenvectors of products of operators from \mathcal{M} . Hence, \tilde{K} contains the root of \mathcal{T} , and therefore, it contains all the nodes $v^{(k)}$. For every r we have $v^{(r)} \notin \text{int } P_{r-1}$. Hence $v^{(r)} \notin \text{int } P_k$ for all $k < r$. Let $g(\cdot)$ be the antinorm defined by the infinite polytope P_k : $g(x) = \sup \{ \lambda \mid \lambda^{-1} x \in P_k \}$. Since a concave function is continuous at every interior point of its domain (see, for instance, [MT]), it follows that g is equivalent to every norm and to every antinorm on the interior cone \tilde{K} . In particular, there are positive constants c_1, c_2 such that

$$c_1 f(x) \leq g(x) \leq c_2 f(x), \quad x \in \tilde{K}, \quad ,$$

where f is an invariant antinorm for $\tilde{\mathcal{M}}$ (Theorem 6). For arbitrary r we have $g(v^{(r)}) \leq 1$. On the other hand, since f is invariant and $f(v_k) = 1$ for all k , we have $f(v) \geq 1$ for every node v of the tree. In particular, $f(v^{(r)}) \geq 1$. Thus, $c_1 \leq g(v^{(r)}) \leq 1$ for all r . Since g is equivalent to each norm on \tilde{K} , we see that the sequence $\{v^{(r)}\}$ is bounded, and hence there is a subsequence $\{v^{(r_i)}\}_{i \in \mathbb{N}}$, $r_1 \geq 1$, that converges to some point $v \in \mathbb{R}^d$. Clearly, $v \in \tilde{K}$ and $c_1 \leq g(v) \leq 1$. For every i we have $v^{(r_i)} = R_i v^{(r_1)}$ and $v^{(r_{i+1})} = C_i v^{(r_i)}$, where $R_i \in \tilde{\mathcal{M}}^{r_i - r_1}$, $C_i \in \tilde{\mathcal{M}}^{r_{i+1} - r_i}$. The sequence $\{v^{(r_i)}\}$ is contained in \tilde{K} , bounded, and separated from zero, hence by Lemma 11 the sequences of operators $\{R_i\}$ and $\{C_i\}$ are both bounded. Therefore, after a passage to subsequences it may be assumed that these two sequences converge to some $R, C \in \text{cl}[\tilde{\mathcal{M}}]$ respectively as $i \rightarrow \infty$ (see the proof of Theorem 4 for the definition of $\text{cl}[\tilde{\mathcal{M}}]$ and $\text{cl}[\tilde{H}]$). We have $Cv = v$. Since $v \in \text{int } \mathbb{R}_+^d$, it follows from Lemma 15 that v is the leading eigenvector of C . Consequently, $\rho(C) = 1$, which, by the domination assumption, implies $C \in \text{cl}[\tilde{H}_j]$ and $v = \lambda v_j$ for some $j = 1, \dots, n$, and $\lambda > 0$. Since $g(v_j) = 1$ and $g(v) \leq 1$, it follows that $\lambda \leq 1$. Thus, $Rv^{(r_1)} = \lambda v_j$. The elements v_j and $v^{(0)} = v_p$ are both from the root, hence there is a product S such that $Sv_j = v^{(1)}$. Taking into account that $v^{(r_1)} = R_1 v^{(1)}$, we obtain $R_1 S R v^{(1)} = \lambda v^{(1)}$. Again invoking Lemma 15, we conclude that $v^{(1)}$ is the leading eigenvector of $R_1 S R$. Hence $\rho(R_1 S R) = \lambda$, and we see that $\lambda = 1$, hence $R_1 S R \in \text{cl}[\tilde{H}_i]$ for some i . This yields $v^{(1)} = \mu v_i$, where v_i is the corresponding vector from the root, $\mu > 0$. Since $g(v_i) = 1$ and $g(v^{(1)}) \leq 1$, we have $\mu \leq 1$. Elements v_i and $v^{(0)} = v_p$ are both from the root, hence $\tilde{A}_p \dots \tilde{A}_i v_i = v_p$, and consequently $\tilde{A}_s \tilde{A}_p \dots \tilde{A}_i v_i = v^{(1)}$ for some $A_s \in \mathcal{M}$. Note that $d_s \neq d_{p+1}$, because the vertex $v^{(1)}$ is not in the

root. Therefore, the product $Q = \tilde{A}_s \tilde{A}_p \cdots \tilde{A}_i$ does not coincide with Π_i , and its length is at most n . Thus, $Qv_i = \mu v_i$, and by Lemma 15 $\rho(Q) = \mu$. Consequently, $\mu = 1$ and v_i is the leading eigenvector of the operator $Q \in \text{cl}[\tilde{\Pi}_i]$. On the other hand the length of the product Q does not exceed n , therefore $Q = \tilde{\Pi}_i$, which is a contradiction. \square

10.3 The 20×20 -matrices A_1, A_2 for the problem of overlap-free words of §8.1 and the proof of Theorem 8.

We write the two matrices A_1, A_2 , associated to the problem discussed in §8.1,

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

respectively.

To give the rigorous proof of the theorem it now suffices to list all the vertices of the extremal polytopes obtained by applying Algorithms **(P)** and **(L)**.

Proof of Theorem 8. Denote $\Pi = A_1 A_2$. To show that

$$\hat{\rho}(A_1, A_2) = [\rho(\Pi)]^{1/2}$$

it suffices to present an extremal polytope P for the operators $\tilde{A}_1 = [\rho(\Pi)]^{-1/2} A_1$ and $\tilde{A}_2 = [\rho(\Pi)]^{-1/2} A_2$. This polytope is $P = \text{co}_-(\{v_i\}_{i=1}^{54})$, where the first vertex v_1 is the leading eigenvector of Π , and the other vertices are

$$\begin{aligned} v_2 &= \tilde{A}_2 v_1, & v_3 &= \tilde{A}_1 v_1, & v_4 &= \tilde{A}_1 v_2, & v_5 &= \tilde{A}_2 v_2, & v_6 &= \tilde{A}_2 v_3, & v_7 &= \tilde{A}_1 v_4, \\ v_8 &= \tilde{A}_1 v_5, & v_9 &= \tilde{A}_1 v_6, & v_{10} &= \tilde{A}_2 v_4, & v_{11} &= \tilde{A}_2 v_5, & v_{12} &= \tilde{A}_2 v_6, & v_{13} &= \tilde{A}_1 v_8, \\ v_{14} &= \tilde{A}_1 v_9, & v_{15} &= \tilde{A}_1 v_{10}, & v_{16} &= \tilde{A}_1 v_{11}, & v_{17} &= \tilde{A}_1 v_{12}, & v_{18} &= \tilde{A}_2 v_7, & v_{19} &= \tilde{A}_2 v_8, \\ v_{20} &= \tilde{A}_2 v_9, & v_{21} &= \tilde{A}_2 v_{10}, & v_{22} &= \tilde{A}_2 v_{11}, & v_{23} &= \tilde{A}_2 v_{12}, & v_{24} &= \tilde{A}_1 v_{13}, & v_{25} &= \tilde{A}_1 v_{14}, \\ v_{26} &= \tilde{A}_1 v_{16}, & v_{27} &= \tilde{A}_1 v_{17}, & v_{28} &= \tilde{A}_1 v_{19}, & v_{29} &= \tilde{A}_1 v_{21}, & v_{30} &= \tilde{A}_1 v_{22}, & v_{31} &= \tilde{A}_2 v_{13}, \\ v_{32} &= \tilde{A}_2 v_{14}, & v_{33} &= \tilde{A}_2 v_{16}, & v_{34} &= \tilde{A}_2 v_{17}, & v_{35} &= \tilde{A}_2 v_{19}, & v_{36} &= \tilde{A}_2 v_{20}, & v_{37} &= \tilde{A}_2 v_{21}, \\ v_{38} &= \tilde{A}_2 v_{22}, & v_{39} &= \tilde{A}_1 v_{25}, & v_{40} &= \tilde{A}_1 v_{26}, & v_{41} &= \tilde{A}_1 v_{27}, & v_{42} &= \tilde{A}_1 v_{30}, & v_{43} &= \tilde{A}_1 v_{32}, \\ v_{44} &= \tilde{A}_1 v_{33}, & v_{45} &= \tilde{A}_2 v_{25}, & v_{46} &= \tilde{A}_2 v_{26}, & v_{47} &= \tilde{A}_2 v_{30}, & v_{48} &= \tilde{A}_2 v_{33}, & v_{49} &= \tilde{A}_1 v_{39}, \\ v_{50} &= \tilde{A}_1 v_{43}, & v_{51} &= \tilde{A}_1 v_{45}, & v_{52} &= \tilde{A}_1 v_{46}, & v_{53} &= \tilde{A}_2 v_{39}, & v_{54} &= \tilde{A}_2 v_{45}. \end{aligned}$$

Now let $\Pi = A_1 A_2^{10}$. To prove that

$$\tilde{\rho}(A_1, A_2) = [\rho(\Pi)]^{1/11}$$

it suffices to present an extremal infinite polytope P for the operators $\tilde{A}_1 = [\rho(\Pi)]^{-1/11} A_1$ and $\tilde{A}_2 = [\rho(\Pi)]^{-1/11} A_2$. This polytope is $P = \text{co}(\{v_i\}_{i=1}^{104}) + K_{\mathcal{H}}$, where the first vertex v_1 is the leading eigenvector of Π , and the other vertices are

$$\begin{aligned} v_2 &= \tilde{A}_2 v_1, & v_3 &= \tilde{A}_2 v_2, & v_4 &= \tilde{A}_2 v_3, & v_5 &= \tilde{A}_2 v_4, & v_6 &= \tilde{A}_2 v_5, & v_7 &= \tilde{A}_2 v_6, \\ v_8 &= \tilde{A}_2 v_7, & v_9 &= \tilde{A}_2 v_8, & v_{10} &= \tilde{A}_2 v_9, & v_{11} &= \tilde{A}_2 v_{10}, & v_{12} &= \tilde{A}_1 v_1, & v_{13} &= \tilde{A}_1 v_2, \\ v_{14} &= \tilde{A}_1 v_3, & v_{15} &= \tilde{A}_1 v_4, & v_{16} &= \tilde{A}_1 v_6, & v_{17} &= \tilde{A}_1 v_7, & v_{18} &= \tilde{A}_1 v_8, & v_{19} &= \tilde{A}_1 v_9, \\ v_{20} &= \tilde{A}_1 v_{10}, & v_{21} &= \tilde{A}_2 v_{11}, & v_{22} &= \tilde{A}_1 v_{12}, & v_{23} &= \tilde{A}_1 v_{13}, & v_{24} &= \tilde{A}_1 v_{18}, & v_{25} &= \tilde{A}_1 v_{19}, \\ v_{26} &= \tilde{A}_1 v_{20}, & v_{27} &= \tilde{A}_1 v_{21}, & v_{28} &= \tilde{A}_2 v_{12}, & v_{29} &= \tilde{A}_2 v_{18}, & v_{30} &= \tilde{A}_2 v_{19}, & v_{31} &= \tilde{A}_2 v_{20}, \\ v_{32} &= \tilde{A}_2 v_{21}, & v_{33} &= \tilde{A}_1 v_{22}, & v_{34} &= \tilde{A}_1 v_{27}, & v_{35} &= \tilde{A}_1 v_{30}, & v_{36} &= \tilde{A}_1 v_{31}, & v_{37} &= \tilde{A}_1 v_{32}, \\ v_{38} &= \tilde{A}_2 v_{27}, & v_{39} &= \tilde{A}_2 v_{30}, & v_{40} &= \tilde{A}_2 v_{31}, & v_{41} &= \tilde{A}_2 v_{32}, & v_{42} &= \tilde{A}_1 v_{34}, & v_{43} &= \tilde{A}_1 v_{37}, \\ v_{44} &= \tilde{A}_1 v_{38}, & v_{45} &= \tilde{A}_1 v_{40}, & v_{46} &= \tilde{A}_1 v_{41}, & v_{47} &= \tilde{A}_2 v_{34}, & v_{48} &= \tilde{A}_2 v_{38}, & v_{49} &= \tilde{A}_2 v_{40}, \\ v_{50} &= \tilde{A}_1 v_{42}, & v_{51} &= \tilde{A}_1 v_{43}, & v_{52} &= \tilde{A}_1 v_{44}, & v_{53} &= \tilde{A}_1 v_{48}, & v_{54} &= \tilde{A}_2 v_{42}, & v_{55} &= \tilde{A}_2 v_{43}, \\ v_{56} &= \tilde{A}_2 v_{44}, & v_{57} &= \tilde{A}_2 v_{47}, & v_{58} &= \tilde{A}_2 v_{48}, & v_{59} &= \tilde{A}_1 v_{50}, & v_{60} &= \tilde{A}_1 v_{52}, & v_{61} &= \tilde{A}_1 v_{54}, \\ v_{62} &= \tilde{A}_2 v_{50}, & v_{63} &= \tilde{A}_2 v_{52}, & v_{64} &= \tilde{A}_2 v_{57}, & v_{65} &= \tilde{A}_1 v_{59}, & v_{66} &= \tilde{A}_1 v_{60}, & v_{67} &= \tilde{A}_1 v_{62}, \\ v_{68} &= \tilde{A}_2 v_{59}, & v_{69} &= \tilde{A}_2 v_{60}, & v_{70} &= \tilde{A}_2 v_{62}, & v_{71} &= \tilde{A}_1 v_{65}, & v_{72} &= \tilde{A}_1 v_{67}, & v_{73} &= \tilde{A}_1 v_{68}, \\ v_{74} &= \tilde{A}_1 v_{69}, & v_{75} &= \tilde{A}_1 v_{70}, & v_{76} &= \tilde{A}_2 v_{65}, & v_{77} &= \tilde{A}_2 v_{67}, & v_{78} &= \tilde{A}_2 v_{68}, & v_{79} &= \tilde{A}_1 v_{71}, \\ v_{80} &= \tilde{A}_1 v_{72}, & v_{81} &= \tilde{A}_1 v_{73}, & v_{82} &= \tilde{A}_1 v_{76}, & v_{83} &= \tilde{A}_1 v_{77}, & v_{84} &= \tilde{A}_1 v_{78}, & v_{85} &= \tilde{A}_2 v_{71}, \\ v_{86} &= \tilde{A}_2 v_{72}, & v_{87} &= \tilde{A}_2 v_{73}, & v_{88} &= \tilde{A}_2 v_{76}, & v_{89} &= \tilde{A}_2 v_{78}, & v_{90} &= \tilde{A}_1 v_{79}, & v_{91} &= \tilde{A}_1 v_{81}, \\ v_{92} &= \tilde{A}_1 v_{82}, & v_{93} &= \tilde{A}_1 v_{84}, & v_{94} &= \tilde{A}_1 v_{85}, & v_{95} &= \tilde{A}_1 v_{87}, & v_{96} &= \tilde{A}_1 v_{88}, & v_{97} &= \tilde{A}_2 v_{79}, \\ v_{98} &= \tilde{A}_2 v_{81}, & v_{99} &= \tilde{A}_2 v_{82}, & v_{100} &= \tilde{A}_2 v_{85}, & v_{101} &= \tilde{A}_2 v_{87}, & v_{102} &= \tilde{A}_2 v_{88}, & v_{103} &= \tilde{A}_1 v_{92}, \\ v_{104} &= \tilde{A}_2 v_{100}. \end{aligned}$$

The proof is completed by routine computations. \square

References

- AG. F. Alizadeh and D. Goldfarb, *Second-order cone programming*, Math. Programming, 95 (2003), No 1, 351.
- ART. E. D. Andersen, C. Roos and T. Terlaky, *On implementing a primal-dual interior-point method for conic quadratic optimization*, Math. Programming, 95 (2003), No 2, 249–277.
- B. N. E. Barabanov, *Lyapunov indicator for discrete inclusions, I–III*, Autom. Remote Control, 49 (1988), No 2, 152–157.
- BW. M. A. Berger and Y. Wang, *Bounded semigroups of matrices*, Linear Alg. Appl., 166 (1992), 21–27.
- Be. J. Berstel, *Growth of repetition-free words – a review*, Theoret. Computer Sc. 340 (2005), No 2, 280–290.
- BJP. V. D. Blondel, R. M. Jungers, and V. Yu. Protasov, *On the complexity of computing the capacity of codes that avoid forbidden difference patterns*, IEEE Trans. Inform. Th., 52 (2006), No 11, 5122–5127.
- BN. V. D. Blondel and Yu. Nesterov, *Computationally efficient approximations of the joint spectral radius*, SIAM J. Matrix Anal., 27 (2005), No 1, 256–272.
- BNT. V. D. Blondel, Y. Nesterov and J. Theys, *On the accuracy of the ellipsoid norm approximation of the joint spectral radius*, Linear Alg. Appl., 394 (2005), 91–107.
- BTV. V. D. Blondel, J. Theys, and A. A. Vladimirov, *An elementary counterexample to the finiteness conjecture*, SIAM J. Matrix Anal. Appl. 24, (2003), No 4, 963–970.

- BT. V. Blondel and J. Tsitsiklis, *Approximating the spectral radius of sets of matrices in the max-algebra is NP-hard*, IEEE Trans. Autom. Control, 45 (2000), No 9, 1762–1765.
- C. J. Cassaigne, *Counting overlap-free binary words*, in STACS 93, Lecture Notes in Comput. Sci. 665, Springer, Berlin, 1993, 216–225.
- CGSZ. A. Cicone, N. Guglielmi, S. Serra-Capizzano, and M. Zennaro, *Finiteness property of pairs of 2×2 sign-matrices via real extremal polytope norms*, Linear Algebra Appl., 432 (2010), No. 2–3, 796–816.
- DL. I. Daubechies and J. Lagarias, *Two-scale difference equations. II. Local regularity, infinite products of matrices and fractals*, SIAM. J. Math. Anal., 23 (1992), No 4, 1031–1079.
- F. S. Finch, private communication, 2008.
- FSB. S. Finch, P. Sebah, and Z.-Q. Bai, *Odd entries in Pascals trinomial triangle*, <http://www.citebase.org/abstract?id=oai:arXiv.org:0802.2654> (2008).
- FV. E. Fornasini, M.E. Valcher, *Stabilizability of discrete-time positive switched systems*, Proceedings of the 49th IEEE Conference on Decision and Control – CDC, (2010), 432–437.
- GKMT. J. Goldwasser, W. Klostermeyer, M. Mays, and G. Trapp, *The density of ones in Pascals rhombus*, Discrete Math., 204 (1999), 231–236.
- G. G. Gripenberg, *Computing the joint spectral radius*, Lin. Alg. Appl., 234 (1996), 43–60.
- GMV. N. Guglielmi, C. Manni and D. Vitale, *Convergence analysis of C^2 Hermite interpolatory subdivision schemes by explicit joint spectral radius formulas*, Lin. Alg. Appl., 434 (2011), 784–902.
- GWZ. N. Guglielmi, F. Wirth, and M. Zennaro, *Complex polytope extremality results for families of matrices*, SIAM J. Matrix Anal. Appl. 27 (2005), No 3, 721–743.
- GZ1. N. Guglielmi and M. Zennaro, *An algorithm for finding extremal polytope norms of matrix families*, Linear Alg. Appl. 428 (2008), No 10, 2265–2282.
- GZ2. N. Guglielmi and M. Zennaro, *Finding extremal complex polytope norms for families of real matrices*, SIAM J. Matrix Anal. Appl., 31 (2009), No 2, 602–620.
- GZ3. N. Guglielmi and M. Zennaro, *Balanced complex polytopes and related vector and matrix norms*, J. Convex Analysis, 14 (2007), 729–766.
- GZ4. N. Guglielmi and M. Zennaro, *On the asymptotic properties of a family of matrices.*, Linear Alg. Appl. 322 (2008), No 1–3, 169–192.
- Gu. L. Gurvits, *Stability of discrete linear inclusions*, Linear Alg. Appl., 231 (1995), 47–85.
- HMR. J. Hechler, B. Mößner, and U. Reif, *C^1 -continuity of the generalized four-point scheme*, Linear Alg. Appl., 430 (2009), No 11–12, 3019–3029.
- HJ. R. Horn and C.R. Johnson, *Matrix analysis*, Cambridge Univ. Press, NY, 1985.
- J. R. M. Jungers, *The Joint Spectral Radius: Theory and Applications*, Vol. 385 in Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin Heidelberg, 2009.
- JP. R. M. Jungers and V. Yu. Protasov, *Counterexamples to the complex polytope extremality conjecture*, SIAM J. Matrix Anal. Appl., 31 (2009), No 2, 404–409.
- JPB1. R. M. Jungers, V. Yu. Protasov, and V.D. Blondel, *Efficient algorithms for deciding the type of growth of products of integer matrices*, Linear Alg. Appl., 428 (2008), No 10, 2296–2312.
- JPB2. R. M. Jungers, V. Yu. Protasov, and V.D. Blondel, *Overlap-free words and spectra of matrices*, Theoret. Computer Sc., 410 (2009), No 38–40, 3670–3684.
- K1. V.S. Kozyakin, *Algebraic unsolvability of problem of absolute stability of desynchronized systems*, Autom. Remote Control, 51 (1990), No 6, 754–759.
- K2. V.S. Kozyakin, *On the computational aspects of the theory of joint spectral radius*, Doklady Math., 80 (2009), No 1, 487–491.
- M. M. Maesumi, *An efficient lower bound for the generalized spectral radius*, Linear Alg. Appl., 240 (1996), 1–7.
- MT. G. G. Magaril-Il'yaev and V.M. Tikhomirov, *Convex analysis: theory and applications*, Transl. from the Russian by Dmitry Chibisov, Translations of Mathematical Monographs 222. Providence, RI, USA, 2001.

- MS. O. Mason and R.N. Shorten, *Quadratic and copositive Lyapunov functions and the stability of positive switched linear systems*, In Proceedings of the American Control Conference (ACC 2007), 657–662.
- PJ. P. A. Parrilo and A. Jadbabaie, *Approximation of the joint spectral radius using sum of squares*, Linear Alg. Appl. 428 (2008), No 10, 2385–2402.
- PWB. E. Plischke, F. Wirth and N. Barabanov, *Duality results for the joint spectral radius and transient behaviour*, Proceedings of the 44 IEEE CDC Conference, Sevilla, Spain, December 12–15 (2005), 2344–2349.
- P1. V. Yu. Protasov, *The joint spectral radius and invariant sets of linear operators*, Fundamentalnaya i prikladnaya matematika, 2 (1996), No 1, 205–231.
- P2. V. Yu. Protasov, *The generalized spectral radius. A geometric approach*, Izvestiya Math., 61 (1997), No 5, 995–1030.
- P3. V. Yu. Protasov, *Asymptotic behaviour of the partition function*, Sb. Math., 191 (2000), No 3–4, 381–414.
- P4. V. Yu. Protasov, *On the regularity of de Rham curves*, Izvestiya Math., 68 (2004), No 3, 27–68.
- P5. V. Yu. Protasov, *Fractal curves and wavelets*, Izvestiya Math., 70 (2006), No 5, 123–162.
- P6. V. Yu. Protasov, *Invariant functionals for random matrices*, Functional Anal. Appl., 44 (2010), No 3, 230–233.
- PJB. V. Yu. Protasov, R. M. Jungers, and V. D. Blondel, *Joint spectral characteristics of matrices: a conic programming approach*, SIAM J. Matrix Anal. Appl., 31 (2010), No 4, 2146–2162.
- R. B. Reznick, *Some binary partition functions*, in Analytic Number Theory: Proceedings of a Conference in Honor of Paul T. Bateman, B. C. Berndt, H. G. Diamond, H. Halberstam, and A. Hildebrand, eds., Boston, Birkhauser, 1990, 451–477.
- RS. G. C. Rota and G. Strang, *A note on the joint spectral radius*, Kon. Nederl. Acad. Wet. Proc. Vol. 63 (1960), 379–381.
- Ru. W. R. Rudin, *Principles of Mathematical Analysis*, Third Edition, McGraw-Hill, 1976.
- Str. G. Strang, *The joint spectral radius*, Commentary by Gilbert Strang, Collected Works of Gian-Carlo Rota, 2000.
- TB. J. N. Tsitsiklis, V. D. Blondel, *The Lyapunov exponent and joint spectral radius of pairs of matrices are hard – when not impossible – to compute and to approximate*, Math. Control Signals Systems, 10 (1997), No 1, 31–40.
- VZ. C. Vagnoni, M. Zennaro, Marino, *The analysis and the representation of balanced complex polytopes in \mathbb{R}^D* , Found. Comput. Math., 9 (2009), No 3, 259–294.
- V. L. Villemoes, *Wavelet analysis of refinement equations*, SIAM J. Math. Anal. 25 (1994), No 5, 1433–1460.
- Va. J. S. Vandergraft, *Spectral properties of matrices which have invariant cones*, SIAM J. Appl. Math. 16 (1968), 1208–1222.

